



HTML'E GİRİŞ

GİRİŞ

HTML, Hyper Text Markup Language kelimelerinin kısaltılmasından oluşan, Türkçe anlamı Zengin Metin İşaretleme Dili olan web sayfalarını oluşturmak için kullanılan standart metin işaretleme dilidir. HTML'in bir programlama dili olmadığı, daha çok betik bir dil olarak kabul edildiği söylenebilir. HTML'in geçmişi incelendiğinde ilk çalışmaların 1980'li yıllara kadar dayandığı görülmektedir. CERN'de bilgi paylaşımını kolaylaştırma amacı doğrultusunda ortaya çıkan HTML, sonraki yıllarda geliştirilmeye devam etmiştir.

- HTML 1.0, 1990'lı yılların başlarında, web tarayıcıları aracılığıyla okunabilen ve erişilebilir bilgileri paylaşma amacıyla yayınlanmıştır. Ancak programcıların çoğu web sitelerini oluşturmakla ilgilenmediğinde büyük bir etki yapmamıştır.
- 1990'lı yılların ortalarına gelindiğinde HTML 1.0'ın tüm özelliklerini ve birkaç ek özelliği içerecek şekilde HTML 2.0 yayınlanmıştır. Ocak 1997'ye kadar web sitelerinin tasarlanması ve yaratılması için standart biçimlendirme dili olarak kaldı ve HTML'in çeşitli temel özelliklerini geliştirdi.
- Sonraki yıllarda Dave Raggett'in yayınlamış olduğu makale ile HTML 3.0 özellikleri tanıtılmaya başlanmıştır. Bu yeni sürüm, web sayfalarını tasarlamada web yöneticileri için daha güçlü özellikler sağlayan, HTML'in geliştirilmiş yeni özelliklerini içeriyordu. Ancak yeni HTML'in bu güçlü özelliklerinin tarayıcıları yavaşlattığı söylenmekteydi.
- Daha sonra yaygın olarak kullanılan ve şu anda dünya çapında piyasaya sürülen ve kullanılan HTML 5.0'dan önce başarılı bir HTML sürümü olan HTML 4.01 gelmektedir. HTML son sürümü olarak HTML 5 ise, 2012 yılında yayınlanan HTML 4.01'in genişletilmiş bir sürümü olarak söylenebilir.

HTML'İN TEMEL ÖZELLİKLERİ

HTML dili etiket yapısından oluşur. Genel olarak <etiket></etiket> şeklinde tanımlanmış alanlar arasında işlemler yapılır. Başka bir ifadeyle işlemler, etiketlerin açılması "<etiket>" ve kapatılması "</etiket>" arasında gerçekleştirilir. HTML'in temel özellikleri şöyle sıralanabilir:

- HTML etiketleri 2 karakter ile sınırlanır: < ve >
- HTML etiketleri çift olarak kullanılır. Örn: <u> Bu metnin altı çizgildir. </u>
- Etiket çiftlerinden ilki başlama etiketi, ikincisi ise bitiş etiketi olarak adlandırılır.
- Başlama ve bitiş etiketleri arasında kalan her şeye element ya da contents, Türkçe olarak ise öge içeriği denir.
- HTML etiketleri büyük ya da küçük harfe duyarlı değildir, yani
 ve
 aynı görevi görür.

HTML'İN TEMEL ETİKETLERİ

Her HTML dosyası içerisinde çok sayıda etiketten oluşmaktadır. Bunları gruplamak gerekirse bunlar; başlık, metin, link, resim, liste, tablo, form gibi başlıklara bölünebilir. Öncelikle bir HTML sayfasında en az üç temel etiket bulunmalıdır: <html>, <head> ve <body>. Örnek bir sayfa şöyle verilebilir:

```
<html>
<head>
<title>Başlığımız</title>
</head>
<body>
Bütün sayfa içeriği buraya yazılır.
</body>
</html>
```

Başlık Etiketleri

HTML’de altı tane başlık etiketi bulunmaktadır. Başlık etiketlerinin yazı boyutları birbirinden farklıdır. <h1> etiketi en büyük yazı boyutuyla başlık yazılır, <h6> etiketi ile ise en küçük başlık yazılır. Örnek:

```
<h6>En küçük başlık</h6>
<h5> . . . </h5>
<h4> . . . </h4>
<h3> . . . </h3>
<h2> . . . </h2>
<h1>En büyük başlık</h1>
```

Metin Etiketleri

HTML’de çok sayıda metin düzenleyici etiket vardır. Bu etiket ile metinlerin şekli, tipi, kalınlığı gibi özellikler değiştirilebilir. Örnek:

```
<p>Paragraf</p>
<br> satır atlatır.
<hr> yatay çizgi çizer.
<pre>Biçimlendirilmemiş metinler </pre>
<strong>Güçlü metin</strong>
<b>Koyu metin </b>
<i>İtalik metin </i>
```

Link (Bağlantı) ve Image (img) Etiketleri

HTML’de link etiketi ile başka sayfalara bağlantılar kurulabilir, image etiketi ile sayfalara resimler eklenebilir. Örnek:

```
<a href="http://atauni.edu.tr">Bu bir bağlantı</a>
<a href="http://atauni.edu.tr"></a>
<a href="mailto:admin@atauni.edu.tr">E-posta gönder </a>
```

Sıralı, Sırasız ve Tanımlamalı Liste Etiketleri

HTML’de elemanları belli bir sıraya göre liste mantığında oluşturmaya yarayan etiketler , ve <dl> etiketleridir.

Sırasız listeye örnek:

```
<ul>
<li>Birinci madde</li>
<li>İkinci madde</li>
</ul>
```

Sıralı listeye örnek:

```
<ol>
<li>Birinci madde</li>
<li>İkinci madde</li>
</ol>
```

Tanımlama listeye örnek:

```
<dl>
<dt>İlk başlık</dt>
<dd>Bilgilendirmesi </dd>
<dt>İkinci başlık</dt>
<dd>Bilgilendirmesi </dd>
</dl>
```

Tablolar (Tables)

HTML sayfalarını daha organize etmek için tablo etiketi kullanılır. Üç temel etiketi vardır: <table>, <tr> ve <td>. Bunlar sırayla tablo, satır ve hücre oluşturmak için kullanılır.

Örnek:

```
<table border="1">
<tr>
<th>Başlık</th>
<th>Başka bir başlık</th>
</tr>
<tr>
<td>Bilgi </td>
<td>Bilgi </td>
</tr>
</table>
```

Çerçeveler (Frames)

Çerçeveler, bir tarayıcı penceresini birden fazla pencereye bölmeye olanak sağlar, böylelikle her bir pencere içerisinde farklı içerikler gösterilmesini sağlar. Örnek:

```
<frameset cols="10%, 90%">
  <frame src="sayfa1.html">
  <frame src="sayfa2.html">
</frameset>
```

Formlar

HTML’de etkileşimli sayfalar yapmak mümkündür. Özellikle kullanıcıdan bilgi almak ve o bilgileri işlemek için formlar kullanılır. Günümüzde, e-ticaret sitelerinden banklara, anketlere kadar birçok içerik formlar ile etkileşimli hâle getirilmektedir. Örnek:

```
<form action="test.php" method="post/get">
  <input type="metin" name="soyisim" size="30" maxlength="50">
  <input type="şifre">
  <input type="checkbox">
  <input type="radio" checked="checked">
  <input type="submit">
  <input type="reset">
  <select>
    <option>Erikler
    <option selected>Üzümler
    <option>Limonlar
  </select>
  <textarea name="aciklama" rows="50" cols="25"></textarea>
</form>
```

Diğer Etiketler

HTML’de çok sayıda etiket vardır. Bu hafta en çok kullanılanları inceledik. Bu etiketlere birkaç tane daha örnek eklemek gerekirse aşağıdakiler verilebilir.

```
<!-- Bu yorum/açıklama-->
```

```
<blockquote>
```

Bu bölümdeki bilgiler başka bir sayfadan alıntılanmıştır.

```
</blockquote>
```



HTML5 İLE GELEN YENİLİKLER

GİRİŞ

HTML5, daha önceki sürümlerin birçok özelliğini barındırmakla beraber yeni oluşturulan ve güncel ihtiyaçlara cevap verebilecek teknolojileri de içermektedir. W3C (World Wide Web Consortium) tarafından önceden duyurulmuş ve modern web tarayıcıları tarafından desteklenecek birçok teknoloji, bu sürüm içerisine dâhil edilmiştir. HTML5'ten önce 1990'lı yılların sonlarında W3C tarafından HTML 4.01 duyurulmuştur. HTML'in bu sürümünden kısa bir süre sonra XML 1.0 yapısı yayınlanmıştır. W3C, HTML dilini XML tabanlı yapmak için bu iki yapıyı birleştirmiş ve 2001 yılında XHTML 1.0 olarak duyurmuştur. W3C, 2003 yılında duyurduğu XForms 1.0 (XHTML Extended Forms) bildirimi ile XHTML yapısını güçlendirmek istemiştir. W3C yazarlarının düşüncesi XHTML yapısının geliştirilmesi ve ek teknolojilerle desteklenmesi şeklindeydi. Bu yapı tarayıcı üreticileri tarafından kabul görmediği için 2004 yılında Apple, Firefox ve Opera tarayıcı üreticileri bir araya gelerek Web Hypertext Application Technology Working Group isimli bir çalışma grubu oluşturdular. Aslında HTML5'in doğuş hikâyesi burada başlamış oldu.

HTML 5'İN TEMEL ÖZELLİKLERİ

HTML, tek başına statik bir web sitesi oluşturmak için yeterlidir. Metin, resim, fotoğraf gibi öğeleri gösteren ve bunları birbirine bağlayan sayfalar arasında etkileşim yaratmak mümkündür. Fakat günümüzde programlama temelli dinamik web sayfaları beklenmekte, daha çok etkileşim talep edilmekte, grafik ve çoklu ortam desteği önemli hâle gelmektedir. Bu nedenle HTML dili üzerinde çalışılarak günümüz teknolojilerini destekleyen HTML5 sürümüne geçiş yapılmıştır. HTML5'in öne çıkan temel özelliklerini sıralayacak olursak: (i) önceki sürümlerle uyumlu olması, (ii) gelişmiş çoklu ortam desteği sunması, (iii) üçüncü parti yazılımlara ve eklentilere ihtiyaç duymadan çoklu ortam ve grafik görüntüleri oynatması, (iv) kompleks web uygulamaları için, konum belirleme, sürükle bırak, yerel depolama gibi yeni API'ler sunması, (v) JavaScript ve CSS teknolojilerinin daha etkili bir şekilde kullanımına imkân tanınması, (vi) 2B ve 3B çizim desteği ile oyunların geliştirilebilmesi.

HTML5 ile Değişen Etiketler

HTML5, HTML'in eski sürümlerinden bağımsız bir sürüm olmayıp daha önceden kullanılan standartlar hâlâ devamlılığını sürdürmektedir. Aslında HTML5 ile HTML'in önceki sürümlerine farklı ve yeni etiketler, elementler eklenerek geliştirilmiştir. HTML5 ile ilk yenilik, DOCTYPE alanına gelen değişiklikle olmuştur. Değişikliğe gidilen bir diğer etiket ise META etiketidir. HTML5'te style ve script etiketlerinin kullanımında da değişikliğe gidilerek, bu etiketler daha sade hâle getirilmiştir. Bütün değişiklikler sonrasında etiketlerin kullanımı son derece basit hâle getirilmiştir.

- `<!DOCTYPE html>`
- `<meta charset="UTF-8">`
- `<style> css kodları </style>`

HTML5 ile Yeni Semantik-Anlamsal Etiketler

HTML5 ile gelen yenilikler incelendiğinde karşımıza birçok yeni etiket çıkmaktadır. Bu etiketlerin özellikle web tasarımı yapanların işlerini kolaylaştırmak üzere üretilmiş olduğu görülmektedir. Bu bölümde öncelikle semantik-anlamsal etiketler incelenmiştir:

- `<aside>` içeriğin ana içerik yanında harici içerik olduğunu belirler.
- `<details>` özet için detay bilgi tanımlar.
- `<figcaption>` `<figure>` etiketiyle beraber kullanılır, başlık belirtir.
- `<figure>` içeriğin resim, şekil gibi görsel olduğunu belirtir.
- `<footer>` sayfanın alt bölüm bilgisi olduğunu belirtir.
- `<header>` sayfanın üst bölüm bilgisi olduğunu belirtir.
- `<mark>` içeriğin vurgulanmış olduğunu belirtir.

- <nav> içeriğin menü değeri olduğunu belirtir.
- <progres> görev ilerleme çubuğu tanımlar.
- <section> içeriğin sayfanın bir bölümü olduğunu belirtir.
- <summary> <details> etiketiyle kullanılır, başlık tanımlar.
- <time> içeriğin tarih/saat olduğunu belirtir.

HTML5 ile Yeni Form Elementleri

HTML5 ile yeni form etiketleri dâhil olmuştur. Bu yeni HTML etiketleriyle form elemanları çeşitlilik göstermektedir. Aşağıda yeni form elemanlarının bazılarından bahsedilmiştir:

- <datalist> veri girişi için önceden tanımlanmış seçeneklerin bir listesini belirtir. Kullanıcılar, veri girdiklerinde önceden tanımlanmış seçeneklerin açılır listelerini görürler.
- <keygen> kullanıcıların kimliğini doğrulamak için güvenli bir yol sağlamaktır. Form gönderildiğinde, biri özel, biri de genel olmak üzere iki anahtar oluşturulur. Özel anahtar yerel olarak saklanır ve genel anahtar sunucuya gönderilir.
- <output> hesaplama veya komut sonucunu temsil eder.

SONUÇ

HTML5 teknolojilerinin kullanımının web sayfalarında yaygınlaştığı görülmektedir. Google başta olmak üzere Facebook, Twitter, LinkedIn, Apple gibi büyük yazılım firmaları tercih etmektedir.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI HTML5 ile Yeni Veri Girişi ve Form Elemanları
ÜNİTE NO 3
YAZAR Doç. Dr. İbrahim Çetin

HTML5 İLE YENİ VERİ GİRİŞİ VE FORM ELEMANLARI

GİRİŞ

Bu bölümde HTML5'in girdi (input) elemanlarından bahsedilecektir. Önceki haftalardan hatırlayacağınız üzere <input> etiketi, <form> ve </form> etiketleri arasında yer alır ve kullanıcılardan farklı tür veri girişi yapacağı alanları tanımlamaktadır. HTML5 ile beraber çok sayıda yeni veri giriş tipi ve form elemanı dâhil olmuştur. Yeni veri giriş tiplerinden sonra bu tiplere kazandırılan yeni özellikler anlatılacaktır.

HTML5 İLE YENİ VERİ GİRİŞ (INPUT) TİPLERİ

HTML5 ile veri girişi tipleri çeşitlilik göstermektedir. Aşağıda yeni veri giriş tiplerinden öne çıkanlar örnek verilerek sıralanmıştır:

- color, renkler için kullanılır, örnek:
<input type="color" name="favoriRenk" />
- email, e-posta için kullanılır, örnek:
<input type="email" name="kullaniciEposta" />
- range, fiyat aralığı vb. yazmak için kullanılır (min, max, step ve value parametreleri alır), örnek:
<input type="range" name="fiyat" min="1" max="10" />
- search, arama girdileri için kullanılır, örnek:
<input type="search" name="arama" />
- tel, telefon yazmak için kullanılır, örnek:
<input type="tel" name="kullaniciTelefonu" />
- url, url girebilmek için kullanılır, örnek:
<input type="url" name="anasayfa" />
- number, sadece sayısal veri girişleri için kullanılır, örnek:
<input type="number" name="miktar" min="1" max="10">
- time, saat için kullanılır, örnek:
<input type="time" name="saat" />
- date, doğumgünü, yıldönümü vb. için kullanılır, örnek:
<input type="date" name="dogumgunu" />
- datetime, date tipine ek olarak saati de kapsayacak şekilde kullanılır, örnek: <input type="datetime" name="dogumgunuSaat" />
- month, ay bilgisini yazmak için kullanılır, örnek:
<input type="month" name="dogumgunuay">
- week, hafta bilgisini yazmak için kullanılır, örnek:
<input type="week" name="hafta" />

HTML5 İLE YENİ FORM ÖZELLİKLERİ

HTML5 ile form elemanlarına yeni özellikler kazandırılmıştır. Özellikle input etiketiyle beraber kullanılan bu özellikler formların kullanılabilirliğini artırmıştır. Aşağıda yeni eklenen özellikler verilmektedir:

- autocomplete, otomatik tamamlama özelliği için kullanılır. Örnek:
<form action="demo.php" autocomplete="on">
 - validate/validate, formun doğrulama yapılması/yapılmaması gereken durumlarda kullanılır.
- Örnek:
- <form action="demo.php" novalidate="novalidate">
- autofocus, sayfa yüklendiğinde otomatik olarak focus/odaklanma özelliği alır. Örnek:
<input type="text" name="fname" autofocus="autofocus" />
 - form, veri girişini, sayfanın başka bir yerinde ki form'a dâhil etmek için kullanılır. Örnek:
Soyisim: <input type="text" name="lastname" form="uyelikformu" />
 - formaction, form'a birden fazla action özelliği kazandırmak için kullanılır. Örnek:
<input type="submit" value="Gönder" />
<input type="submit" formaction="demo2.php" value="Gönder 2" />
 - formmethod, form'a birden fazla method özelliği kazandırmak için kullanılır. Örnek:
<input type="submit" value="Gönder 1" />
<input type="submit" formmethod="post" formaction="demoPost.php" value="Gönder 2" />
 - formnovalidate, form elemanları kontrol edilmeden gönderimi için kullanılır. Örnek:
<input type="submit" value="Gönder 1" />
<input type="submit" formnovalidate="formnovalidate" value="Gönder 2" />
 - formtarget, form verileri gönderildikten sonra nerede görüntüleneceğini gösteren bir isim ya da anahtar sözcük belirlemek için kullanılır. Örnek:
<input type="submit" value="Gönder 1" />
<input type="submit" formtarget="_blank" value="Gönder 2" />
 - min ve max özellikleri number, range, date, datetime-local, month, time ve week gibi girdilerin en küçük ve en büyük değerlerini belirlemek için kullanılır. Örnek:
<input type="date" name="gun" max="2019-12-31">

<input type="number" name="miktar" min="1" max="10">

 - multiple, eposta ve dosya seçimi gibi form elemanlarına birden fazla değer girilmesi için kullanılır. Örnek:
<input type="file" name="resim" multiple>
 - pattern, text, search, url, tel, email, ve password gibi veri girişlerinin kontrollü yapılması sağlayabilmek için kullanılır. Örnek:
<input type="text" name="kod" pattern="[A-Za-z]{2}" title="İKİ HARFLİK">
 - placeholder, bir giriş alanına beklenen değerini açıklayan kısa bilgi vermek için kullanılır. Örnek:
<input type="text" name="AD" placeholder="İsminizi giriniz">
 - required, form gönderilmeden önce istenen girdilerin doldurulmasını zorunlu kılmak için kullanılır. Örnek:
<input type="text" name="ad" required>
 - step, genellikle min ve max özellikleriyle birlikte kullanılarak geçerli sayı aralığında veri girişi yapılmasını sağlamak için kullanılır. Örnek:
<input type="number" name="veri" step="3" min="-30" max="30">
 - readonly, sadece okumaya izin verilir, değişikliğe veya yeni veri girişine izin vermek için kullanılır. Örnek:
<input type="text" name="ulke" value="Türkiye" readonly>

SONUÇ

Bu bölümde HTML5'in yeni giriş (input) elemanları anlatılmıştır. HTML5 ile beraber çok sayıda yeni veri girişi tipi ve form elemanı dâhil olmuştur. Böylelikle etkileşimli formlar hazırlanabilmektedir.



HTML5 İLE İLERİ TEKNOLOJİLER

GİRİŞ

HTML5'in en büyük yeniliklerinden biri, ileri teknoloji olarak söyleyebileceğimiz, web sayfalarındaki birçok yeni özelliğe kolaylıkla adapte olabilmesidir. Çoklu ortam desteği, grafik çizimler, konum belirleme özelliği, sürükle bırak uygulamaları, önbellekleme gibi güncel teknolojiler artık HTML5 ile hem kolay hem de hızlıca yapılabilmektedir. HTML5'in yeni elementleriyle birkaç satırda video ve ses dosyaları sayfalardan paylaşılabilir, JavaScript ve CSS desteği ile istenilen çizimlerle oyunlar yapılabilir, haritalar üzerinde konumlar belirlenebilir, sayfalar çevrimdışı durumunda da görülebilir.

HTML5 İLE ÇOKLU ORTAM

HTML5 ile çoklu ortam dosyaları ses, video gibi, Silverlight ya da Flash gibi eklentilere ihtiyaç duymadan web sayfalarına eklenebilmektedir. HTML5 ile gelen bu yeni standardın yaygın olarak bilinen çoklu ortam etiketleri şunlardır: video, audio, embed ve object.

- <video> etiketi ile bir video web sayfasında yayınlanabilmektedir.
- <audio> etiketiyle de ses ve şarkılar web sayfalarından yayınlanabilir.
- <embed> ve <object> etiketleriyle de web sayfalarına ses, video ve animasyon eklenebilmektedir.

HTML5 İLE GRAFİK

HTML5 ile grafik özelliklerinin kuvvetlenmesi için <canvas> ve <svg> etiketleri geliştirilmiştir.

- <canvas> etiketi ile sayfada bir tuval alanı (dikdörtgen alanı) oluşturur. Bu etiket yardımıyla grafikler çizilebilir, fotoğraf kompozisyonları yapılabilir veya animasyonlar hazırlanabilir.
- <svg> etiketi ile vektör temelli grafikler çizilebilir ve grafikler yeniden boyutlandırılırken veya büyütülüp küçültülürken kaliteleri düşmez.

HTML5 İLE KONUM BELİRLEME

Geolocation API ile konum bilgilerini almak ve alınan bilgileri harita üzerinde göstermek mümkündür. Bu API'yi kullanabilmek için HTML5 ile birlikte JavaScript de kullanmak gerekir. Özellikle navigator.geolocation ve getCurrentPosition JavaScript metotları ile konum bilgileri alınmaktadır.

HTML5 İLE SÜRÜKLE BIRAK

HTML5 ile sürüle bırak, yani metni veya resmi başka bir alana fare yardımıyla taşımak daha kolay hâle gelmiştir ve tüm nesnelere uygulanabilmektedir. Bu özelliği kullanabilmek için de CSS ve JavaScript'in beraber kullanılması gerekir. <div> etiketi, ondragover ve ondragstart gibi fonksiyonlar sık kullanılan komutlardandır.

HTML5 İLE UYGULAMA ÖNBELLEKLEME

Web sayfaların önbelleğe (cache) atılıp, çevrimdışı durumunda da çalışabilir olmasına HTML5'te "Uygulama Önbellekleme (Application Cache)" denir. Web uygulamasının bir çevrimdışı sürümünü yapmak için bir manifest dosyası oluşturmak ve manifest sınıfını (attribute) belgenin HTML etiketine eklemek yeterlidir. HTML etiketine <html manifest="uygulama.appcache"> yazılarak ekleme yapılabilir.

HTML5 İLE GELEN DİĞER API'LER

HTML5 ile beraber çok sayıda API desteği gelmiştir. Birkaç API desteği ile ilgili bilgiler şöyle verilebilir:

- Yerel Depolama Alanı (Local Web Storage): HTML5 ile beraber yerel depolama birimleri ile kullanıcı tarayıcısına yerel olarak veri depolanabilmektedir. İki tip web depolama versiyonu vardır. Birincisi, Local Storage ile kaydedilen bilgiler, belirtilmeyen herhangi bir tarihe kadar saklanabilmektedir. Diğeri, Session Storage ile istenilen bilgiler sadece bir oturum (session) için

saklanmaktadır.

- Web (Workers) İşçileri: Web sayfalarında JavaScriptler, tek işte çalıştırılacak şekilde tasarlanmıştır, yani birden çok komut aynı anda çalıştırılmaz. Web işçisi, sayfanın performansını etkilemeden diğer script'lerden bağımsız olarak arka planda çalışan bir JavaScript kodudur.
- SSE (Server-Sent Events): HTML5 ile Server-Sent Olayları, web sayfasında kullanılmak üzere sunucudan güncellemeleri almak için kullanılır.

HTML5 İLE KALDIRILAN ETİKETLER

HTML5 ile beraber bazı etiketler artık kullanılmamaya başlanmıştır. Kaldırılan etiketler şunlardır:

<acronym>, <applet>, <basefont>, <big>, <center>, <tt>, , <frame>, <frameset>, <noframes>, <strike>, <dir>

SONUÇ

Sonuç olarak HTML5 ile ilgili gelişmeleri incelediğimizde, piyasaya sürüldüğünden itibaren çok ses getiren bir teknoloji olduğu görülmektedir. HTML5'in bizlere sağladığı temel kazanımları 3 maddede özetleyecek olursak: daha fazla işlevsellik, daha fazla semantik, daha fazla etkileşim.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI CSS ve Metin Stilleri
ÜNİTE NO 5
YAZAR Prof. Ercan TOP

CSS NEDİR

CSS, "Cascading Style Sheets" kelimesinin baş harflerinden oluşmaktadır. CSS (Türkçedeki karşılığı Basamaklı Stil Sayfaları), tek tek sayfalar için stil etiketleri (tag) yazma zorluğunu ve karmaşasını azaltmak için yaratılmıştır. CSS, sayfa şekillendirmeleri için kullanılmaktadır. CSS, sayfa stilini ve düzenini tek bir konumda tanımlamaya olanak tanır, böylece onlar genel olarak tüm .htm dosyalarına uygulanabilir.

CSS Kural Yapısı

Stil sayfası, sayfa öğelerinin nasıl sunulması gerektiğini açıklayan bir veya daha fazla kuraldan oluşur. Her kuralın seçici ve bildirim bloğu olmak üzere iki temel bölümü vardır. Kuralın sol tarafında, kuralın uygulanması gereken belgenin bölümlerini seçen seçici vardır. Kuralın sağ tarafında bildirim bloğu vardır. Bir bildirim bloğu bir veya daha fazla bildirimden oluşabilir

CSS SÜRÜMLERİ

World Wide Web konsorsiyumunun tavsiyesi olarak ilki 1996'da ortaya çıkmıştır. Üç tane sürümü vardır, genel kabul görmüş standart bir CSS4 spesifikasyon yoktur.

CSS TANIMLAMA YÖNTEMLERİ

Satır İçi

Satır içi stil tanımlama, etiketin kendi içinde yapılır. Stil bilgisi, stil özelliği yoluyla tek bir öğe için belirtilebilir. Sayfadaki diğer etiketleri etkilemez. Sayfanın her yerinde istenildiği kadar tekrarlanabilir.

Belge Düzeyinde

Belge düzeyinde stil tanımlama aynı belgenin başlığında (<head> etiketinin içinde) yapılır. Stil tanımlamaları genellikle <head> etiketinin içinde bulunsa da bu zorunlu bir kullanım şekli değildir.

Dış Stil Sayfaları

Stiller ayrı bir dosyada saklanabilir. Dış stil sayfalarında, stil bildirimlerini içeren farklı bir dosya oluşturulur ve bunun ardından ana html dosyasına bağlanır. Bu oluşturulan yeni belge ".css" uzantısı kullanılarak kaydedilmelidir.

CSS SEÇİCİLERİ

Basit Seçici

Basit seçici, etiketin seçici olarak kullanıldığı temel stil tanımlama türüdür.

Class (Sınıf) Seçicisi

Sınıf seçicisi, aynı etiketin iki veya daha fazla tekrarına farklı stiller uygulamamız gerektiğinde kullanılabilir.

Genel Seçici

Aynı stil tanımlamaları iki veya daha fazla farklı etikete kullanılmak istendiğinde, genel seçici kullanılabilir.

ID Seçici

Sadece tek bir etikette kullanılabilir ve iki etikete verilemeyen özel seçicilerdir.

CSS VE METİN

Fonts (Yazı Tipi)

CSS kullanılarak html belgelerindeki metnin varsayılan yazı tipleri, görünümü ve biçimi değiştirilebilir. Yazı tipleri web sayfalarının en temel özelliklerinden birisidir. Yazı tipleri, web tasarımının ve Web'de metinlerin nasıl gösterildiğinin ayrılmaz bir parçasıdır.

Font-Family (Yazı Tipi Ailesi)

Metnin fontunu değiştirmek için font-family özelliği kullanılır. font-family özelliği bildiriminde virgül ayırıcısı kullanılarak birden çok yazı tipi bildirilebilir.

Font-Size (Yazı Tipi Boyutu)

Yazı tipi boyutunu tanımlamak için font-size özelliği kullanılır. Metin boyutları, piksel (px), punto (pt)

veya göreceli boyutlardan “em” veya yüzde olmak üzere dört şekilde tanımlanabilir.

Text-Align (Metin Hizalama)

Metni sağa, sola veya merkeze hizalamak için text-align özelliği kullanılır. text-align özelliği ile hangi yönün metnin başlangıcı olarak kabul edileceğini ve hangisinin bitiş olarak kabul edileceğini belirlenir.

Text-Decoration (Metin Dekorasyon)

Metin dekorasyon özelliği, metnin altından çizgi çizmek (underline), metnin üstünden çizgi çizmek (overline) ve metnin üzerinden çizgi çizmek (line-through) için kullanılabilir.

Font-Weight (Metin Koyuluğu)

font-weight özelliği, metnin ağırlığını (koyuluğunu) tanımlamak için kullanılabilir.

Font-Style (Yazı Tipi Özelliği)

Yazı tipi stili özelliği, yazı tipi stilini normalden italik ya da eğik olarak değiştirmek için kullanılır.

Text-Transform (Metin Dönüştürme)

text-transform özelliği, metni büyük veya küçük harflere dönüştürmek veya kelimelerin ilk harflerini büyük harfe dönüştürmek için kullanılır.

Line-Height (Satır Arası Yükseklik)

Satırlar arasındaki yüksekliği belirlemek için line-height özelliği kullanılır.

Bağlantı (Link) Kullanımı ve Özellikleri

Linklerin farklı durumları vardır, yani bir web sayfasında onlarla etkileşime girildiğinde duruma adapte olurlar. Varsayılan bağlantı stili değiştirilmek istendiğinde, dikkate alınması gereken üç ek (hover, active ve visited) bağlantı durumu vardır.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI CSS'te Renk Tanımı ve Tablo Düzenleme
ÜNİTE NO 6
YAZAR Prof. Ercan TOP

RENK

Renkler, ilgili herhangi bir CSS kodu kullanılarak web sitesine veya blog'a uygulanabilirler. Metin rengini ayarlamak için color, arka plan rengini ayarlamak için background-color ve kenarlık rengini ayarlamak için border-color özelliği kullanılmaktadır. Bütün özelliklerde aynı renk bildirim çeşitleri kullanılabilir. Color bildirimi için 3 farklı yöntem vardır.

Renk Adı

Renk adı doğrudan beyan edilerek renk bildirilebilir. Adlandırılmış renkler İngilizce isimleriyle bildirilmek zorundadırlar ve büyük / küçük harfe duyarlı değillerdir.

Onaltılık Kod (hexadecimal code)

Renk, her 2 hanenin kırmızı, yeşil, mavi renklerini temsil ettiği 6 basamaklı onaltılık sayı kullanılarak bildirilebilir. Onaltılık kod bayt değerleri, bir rengin en düşük yoğunluğu olan 00 ile en yüksek yoğunluğu temsil eden FF arasındadır.

RGB Kodu

İstenilen rengi elde etmek için ne kadar kırmızı, yeşil ve mavi rengin kullanıldığı bildirilen RGB yöntemi kullanılarak da renk seçimi yapılabilir. Beyanlar rgb ile başlayıp parantez içinde virgüllerle ayrılmış üç sayı bloğundan oluşmaktadır. Her bir sayı blokunun alabileceği değerler 0 ile 255 arasındadır.

Renk (color)

Renk özelliği, metnin ön plan rengini değiştirmek için kullanılır. Color özelliği bütün tarayıcıların ilk sürümünden itibaren desteklenmektedir.

Arka Plan (background)

Arka plan renklerini bildirmek için background-color özelliği kullanılır.

SPAN

HTML etiketi, satır içi öğeler ve içerik için genel bir satır içi kapsayıcıdır. Öğeleri stillendirme amacıyla gruplamak için kullanılır (sınıf veya kimlik özniteliklerini kullanarak), başka hiçbir etiket mevcut olmadığında kullanmak için daha iyi bir yoldur.

TABLO DÜZENLEME

Kenar (border)

CSS'de tablo kenarlığını belirtmek için border özelliği kullanılır. Border özelliğinin, border-width (kenarlık genişliği), border-style (kenarlık stili) ve border-color (kenarlık rengi) olmak üzere üç alt özelliği bulunmaktadır. Üç alt özelliği sadece border özelliği ile tek seferde tanımlamak mümkündür.

Kenar Daraltma (border-collapse)

Tablo kenarlıklarının tek bir kenarlığa daraltılması veya standart HTML'deki gibi mi ayrılacağını belirler. Tablo hücreleri etrafındaki kenarlık ile olan mesafeyi düzenlemeye olanak sağlar. Border-collapse, collapse (daralt) ve separate (ayrık) değerlerini alabilir.

Yükseklik ve Genişlik (height ve width)

Tablonun ve tablodaki hücrelerin genişlik ve yüksekliği, height ve width özellikleri tarafından tanımlanır. Yüksekliği ve genişliği bildirmek için yüzdelik değerler (% , em) veya tam değerler (piksel, pt, cm) kullanılabilir.

Dolgu (padding)

Padding özelliği, <td> ve <th> etiketlerinin içerisindeki içeriğe fazladan bir dolgu alanı ekler. Bu, tablonun etrafındaki alanı etkili bir şekilde kontrol etmek için kullanılabilir. Bir etiketin dolgusu (padding), etiketin içeriği ile sınırı arasındaki boşluktur. Padding özelliği, aslında padding-top, padding-right, padding-bottom ve padding-left özellikleri için bir kısa yol özelliğidir. Padding değerleri bütün yönler için bir defada bütün olarak verilmek zorunda değildir. Her bir yön için ayrı ayrı stil bildirimi yapılabilir. Stil tanımında bütün yönler için padding bildirimi yapılmış olmazsa, bildirim yapılmayan yönler için varsayılan değerler kullanılır.

Kenar boşlukları (margin)

CSS margin özelliđi, etiketlerin etrafında, tanımlanmış sınırların dışında boşluk oluşturmak için kullanılır. Margin özelliđi ile kenar boşlukları üzerinde tam kontrol kullanıcıda olmaktadır. Bir etiketin her bir tarafı için kenar boşluđunu ayarlama özellikleri bulunmaktadır. Margin özelliđinde, padding özelliđinde olduđu gibi tek bildirimle bütün yönlerdeki kenar boşluk miktarını belirlemek, kenar boşluk miktarını ayrı ayrı belirlemek veya sadece tek bir yöndeki kenar boşluk miktarını belirlemek mümkündür. Margin özelliđi için stil tanımlamada farklı sayıda deđerler de kullanılabilir.

Margin ve Padding

Padding (dolgu), bir eleman içinde fazladan boşluk yaratırken, margin (kenar boşluđu) bir elemanın etrafında fazladan boşluk yaratır.

Tablo Rengi

Tabloyu görsel olarak daha çekici hale getirmek için tablodaki arka plan rengi ve metnin rengi deđiştirebilir. Arka plan rengi için background-color özelliđi ve metin rengi için color özelliđi kullanılır.

Hover Seçici

Hover seçici, fare üzerine getirildiğinde belirli bir hücreyi veya sırayı vurgulamayı sağlar.

Şeritli Tablo

Şeritli tablo oluşturmak için “nth-child” seçicisi kullanılabilir. Tek veya çift numaralı tablo sırası seçilerek ikisinden birinin veya ikisinin de arka plan rengi deđiştirebilir. Büyük tablolar olduđuunda, şeritli tablo kullanmak iyi bir uygulamadır, izleyicinin aradıđı içeriđi kolayca aramasında ve satırları gözüyle zorlanmadan takip etmesinde yardımcı olur. Deđer olarak sayı (tablonun kaçınıcı satırı olduđunu belirtmek için), odd (bütün tek numaralı satırları belirtmek için), even (bütün çift numaralı satırları belirtmek için) ve n (tam sayıları- çeşitli şekillerde formüller oluşturmak için) kullanılabilir.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI Liste ve Görsellerde CSS Stilleri
ÜNİTE NO 7
YAZAR Prof. Ercan TOP

LİSTS (LİSTELER)

HTML’de iki ana tür liste vardır:

- `` (unordered list - sıralanmamış listeler): Liste öğeleri imlerle işaretlenir.
- `` (ordered list - sıralı listeler): Liste öğeleri sayılarla veya harflerle işaretlenir.

list-style-type

Sıralı ve sıralanmamış listeler için çeşitli seçenekler ve işaretçiler bulunmaktadır. List-style-type özelliği, liste öğesi işaretleyicisinin türünü belirtir. Sıralanmamış listeler için list-style-type özelliğinin alabileceği değerler; disc, square, circle ve none olarak belirlenmiştir. Sıralanmış listeler için list-style-type özelliğinin alabileceği bazı değerler; decimal, decimal-leading-zero, lower-roman, upper-roman, lower-greek, upper-greek, lower-alpha, upper-alpha, lower-latin, upper-latin olabilir.

list-style-image

Sıralanmamış listeler için liste öğesi işaretçisi olarak bir görüntü tanımlamaya olanak sağlar.

list-style-position

İşaretçilerin konumlandırılmasına olanak sağlar. Bu özellik içeride (inside) ve dışarıda (outside) değerleri bildirilerek kullanılabilir.

li:hover

Fare, üzerine getirildiğinde, farenin üzerinde olduğu liste ögesini vurgulamayı sağlar. Örnek olarak imleç, sıralanmamış listede liste ögesinin üzerine geldiğinde ögenin arka plan rengini değiştiren stil bildirimi “li:hover{background: #88ee55;} olarak uygulanabilir.

GÖRÜNTÜLER (IMAGES)

CSS, sayfalarda kullanılan görüntülerin nasıl görüntüleneceğinin ayarlanmasına olanak sağlar. Tüm görsellere stiller ayarlayarak, görseller için standart bir görünüm yaratılabilir. Bu sayede, sayfalar arasında tutarlık sağlanarak sayfa ziyaretçilerine daha profesyonel bir izlenim yaratılabilir.

Border

Görüntülere kenarlık eklemek için border özelliği kullanılabilir. Border özelliği ile kenar boşlukları ve/veya dolgu özelliği kullanmak görüntülerin sayfalarda daha etkili bir şekilde yerleştirilmesi için işe yarayabilmektedir.

border-image

border-image (kenarlık resmi) özelliği ile bir ögenin etrafındaki kenarlık olarak kullanılacak bir görüntü ayarlanabilmektedir. border-image özelliği, bir elemanın çevresindeki normal kenarlık yerine kullanılacak bir resim belirlemeye olanak sağlamaktadır. Özellik tanımlaması kenarlık olarak kullanılacak görüntü, resmin nereden dilimleneceği ve orta bölümlerin tekrarlanacağını veya uzatılacağını tanımlayan üç bölümden oluşmaktadır.

border-radius

Yuvarlak (oval) köşeli görüntüler oluşturmak için border-radius özelliği kullanılabilir. Kenarlık yarıçapı özelliği, ögenin köşelerinin yarıçapını tanımlar. Bir başka ifade ile bu özellik, öğelere yuvarlatılmış köşeler eklemeye olanak sağlar. Bütün köşeler için tek değer tanımlanabildiği gibi ayrı ayrı her bir köşe için de yarıçap tanımlanabilmektedir. Aynı görselin farklı köşeleri için farklı yuvarlama değerleri de tanımlanabilmektedir. Border-radius özelliği bildiriminde bir, iki, üç ve dört değer de kullanılabilir. border-radius özelliği için değer olarak % oranı da kullanılabilir. Sayısal değerlerde olduğu gibi bir, iki, üç ve dört değerli tanımlamalar yapılabilir. border-radius özelliğinin yüzdelik değeri 100% olarak tanımlandığında, tam yuvarlak veya elips bir görüntü elde edilebilir.

Opacity

opacity özelliği kullanarak bir görüntünün saydamlık derecesi tanımlanabilir. opacity özelliği için 0.0 ile 1.0 arasında değerler tanımlanabilir. Opaklık özelliğinde kullanılan değerlerden, 1 görselin hiç şeffaf olmadığını, 0,5 görselin %50 şeffaf olduğunu ve 0 görselin tamamen şeffaf olduğunu tanımlar.

Boyutlar

Görüntülerin yüksekliğini (height) ve genişliğini (width) değiştirmek veya ayarlamak için stil tanımlanabilmektedir. İndirme hızları nedeniyle görüntü boyutlarını ayarlamak için tarayıcıyı kullanmak iyi bir fikir değildir ama yine de sayfaların düzenli görüntülenmesi için sıklıkla kullanılmaktadır. Görüntüler yeniden boyutlandırıldığında, en iyi sonuçları elde etmek için, sadece bir boyutu (yükseklik veya genişlik) yeniden boyutlandırmak gerekir. Bu, görüntünün en boy oranını korumasını sağlar ve böylece görüntünün tarayıcıda değişik görünmesi engellenebilir.

background-image

CSS, görsellerle süslü arka planlar oluşturmayı kolaylaştırır. Tüm sayfaya veya yalnızca belirli bir öğeye arka planlar eklenebilir. Sayfada bir arka plan görseli oluşturmak background-image özelliği ile oldukça kolay bir işlemdir. “background-attachment” özelliği, scroll, fixed ve local değerleri alabilir ve arka plan görüntüsünün sayfanın geri kalanıyla kaydırılmasını veya sabitlenmesini belirler. “background-repeat” özelliği kullanılarak belirtilen görselin yalnızca yatayda, dikeyde veya sadece bir defa döşenmesi sağlanabilir. “background-repeat” özelliğinin alabileceği değerler repeat, repeat-x, repeat-y, no-repeat, space ve round olarak belirlenmiştir.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI CSS'te Kutu Modeli, Kalıtsallık ve Öncelikler
ÜNİTE NO 8
YAZAR Prof. Ercan TOP

UZUN ÖZET

CSS KUTU MODELİ

CSS'te, tasarım ve düzen hakkında konuşulurken "kutu modeli" terimi kullanılmaktadır. Tüm HTML öğeleri kutular olarak kabul edilebilir. Kutu modeli, HTML öğelerinin etrafına kenarlık eklenmesine ve elemanlar arasında boşluk tanımlanmasına olanak sağlar. Kutu modeli kenar boşlukları, kenarlıklar, dolgular ve içerikten oluşmaktadır. Bir öğenin genişlik ve yükseklik özellikleri CSS ile ayarlandığında, yalnızca içerik alanının genişliği ve yüksekliği ayarlanmış olur. Bir öğenin tam boyutunu hesaplamak için, dolgu, kenarlıklar ve kenar boşluklarının da eklenmesi gerekmektedir.

Div

<div> etiketi bir web sayfasını düzenlemek için CSS ile birlikte çok sık kullanılmaktadır. HTML'deki kutu modeli teriminin iyi bir örneğini teşkil etmektedir. <div> etiketi genellikle diğer HTML öğelerinin CSS kullanılarak stillendirilmesi ile belirli görevleri gerçekleştirmek için bir kapsayıcı olarak kullanılmaktadır. Varsayılan olarak, tarayıcılar her zaman <div> ögesinden önce ve sonra bir satır sonu vermektedir.

Float ve Clear

CSS float özelliği, bir öğenin ekranda nasıl durması gerektiğini belirtmek için kullanılır. float özelliği, örneğin içeriği yerleştirmek ve biçimlendirmek için kullanılabilir. "float" blok seviye elemanlarını üst üste yerleştirmek yerine yan yana yerleştirmeye olanak sağlar. CSS clear özelliği, hangi öğelerin temizlenen öğenin yanında ve hangi tarafında kalacağını belirlemek için kullanılmaktadır. Clear özelliği genelde bir ögede float özelliği ile kullanılmaktadır.

Position

Position özelliği, bir öge için kullanılan konumlandırma yönteminin türünü belirtmektedir. Özellik, bir içeriğin sayfada nerede görüntüleneceğini tanımlamak için kullanılmaktadır. Öğeler top (üst), right (sağ), bottom (alt), left (sol) ve z-index (z-indeks) değerleri kullanılarak konumlandırılır. Ancak, "position" özelliği öncelikle ayarlanmadıkça bu değerler çalışmaz. Ayrıca değerler, "position" özelliğine bağlı olarak farklı çalışırlar.

Static

HTML öğeleri varsayılan olarak statik olarak konumlandırılmıştır. Statik konumlandırılmış öğeler, üst, alt, sol ve sağ değerlerinden etkilenmezler. Position değeri "static" bir öğenin içeriği farklı şekilde konumlandırılmaz, içerik sayfanın normal akışına göre yerleştirilir.

Relative

relative seçeneği, sayfanın normal akışındaki konumuna göre öğeyi yerleştirmek için kullanılmaktadır. Relative ve statik olmayan tüm konum değerleri için öğenin varsayılan konumu değiştirilmektedir. Fakat sadece position özelliği için relative değerini bildirmek yeterli değildir, aynı zamanda elementin koordinatlarının yardımcı özelliklerle de ayarlanması gerekmektedir.

Fixed

Fixed seçeneği, sayfa aşağı, yukarı veya yanlara doğru kaydırıldığında içeriğin her zaman görünür olmasını sağlamak için kullanılmaktadır. Belirtilen öğeyi görünüm alanında sabitleştirir, sayfayı kaydırmak, sabit konumlandırılmış öğeyi etkilemez. Sabit bir öge, sayfada normalde yerleştirileceği yerde (içerik akışında olması gereken yerde) herhangi bir boşluk bırakmaz.

Absolute

absolute seçeneğindeki içerik, üst öğeye göre konumlandırılmaktadır. Üst öğenin başlangıç noktasına (sol üst köşe) göre otomatik olarak yerleştirilmektedir. "absolute" olarak tanımlanan öğeler normal belge akışından kaldırılır. "absolute" olarak tanımlanan öge, belge akışından tamamen kaldırdığından dolayı, diğer öğeler bu durumdan etkilenmekte ve öge web sayfasından tamamen kaldırılmış gibi işleme devam edilmektedir.

Sticky

sticky kullanımı ile kullanıcının kaydırma konumuna göre konumlandırma yapılabilir. “Sticky” olarak tanımlanan öge, kaydırma konumuna bağlı olarak göreceli (relative) ve sabit (fixed) değer arasında geçiş yapabilmektedir. Oluşturulan öge kaydırma çubuğu kendi konumuna gelene kadar “relative” (göreceli) değerindeki gibi konumlandırılır, daha sonra değer “fixed” olarak tanımlanmış gibi sayfada gösterilmektedir.

Z-index

Üst üste binen 2 veya daha fazla içerik olduğunda, üst üste binen içeriğin hangisinin üstte görüntüleneceğini tanımlamak için z-index özelliği kullanılmaktadır. z-index özelliğinin uygulanabilmesi için öncelikle öğelere konumlandırma (position) değeri atanmış olması gerekmektedir.

CSS’DE ÖNCELİKLER

Basamaklama (cascading), CSS’te kural kümesinin adıdır. Oluşturulan bildirimlerde ortaya çıkan çatışmaların nasıl çözüldüğünü belirlemektedir ve dilin nasıl çalıştığının temel bir parçasıdır. CSS bildiriminde kullanılan öncelik sıralarını incelemek, sayfalarda yapılan stil düzenlemelerinin istenilen şekilde görünmesini sağlamak için önemlidir.

Satır içi stil / belge düzeyinde stil

Satır içi stilde bir etiket, sınıf veya ID için özellikler tanımlanır. Bir html dosyasında belge düzeyinde stillere göre öncelikle baskın olan kısım satır içi stillerdir.

Sayfaya gömülü stil / dış stil dosyaları (CSS)

Bir html dosyasında sayfaya gömülü stiller, CSS dosyasındaki stillere göre baskın olan kısımdır.

Dosyada tanımlı stil olması / kullanıcı stili

Kullanıcı stilinde HTML dosyasında herhangi bir stil bildirimi yapılmamışsa ve kullanıcı tarayıcıda stil tanımlamalarında değişiklik yapmışsa, tarayıcıda kullanıcının yapmış olduğu bildiriler etkin olur.

Kullanıcı stili / tarayıcı stili

Tarayıcı stilinde HTML dosyasında herhangi bir stil bildirimi yoktur ve kullanıcı da tarayıcının stil bildirimlerinde değişiklik yapmamıştır, bu dosyanın gösteriminde tarayıcı tarafından varsayılan stil ayarları kullanılır.

CSS’DE KALITSALLIK (INHERITANCE)

HTML öğeleri bir soyağacında olduğu gibi bir bağ ile birbirine bağlıdır. Aynı aile içinde kalıtsal değerlerin büyükten küçüğe geçtiği gibi HTML öğeleri arasında da CSS özellikleri kalıtsal olarak alt etiketleri (çocuk etiket) etkiler. Kalıtsallık CSS kullanımında bir bildirimin tek tek bütün etiketlere uygulanması yerine ata etikete uygulayarak alt (çocuk) etiketlere de otomatik olarak uygulanmasını sağlar. Bir ögenin stilleri alabilmesinin bir yolu da mirastır (kalıttır). Bütün özellikler miras alınmaz, margin (kenar), padding (dolgulama), background (arka plan rengi), width (genişlik) gibi bazı özelliklerde kalıtsallık yoktur.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI Javascript ile Kodlamaya Giriş
ÜNİTE NO 9
YAZAR Dr. Öğr. Üyesi HALİL ERSOY

JAVASCRIPT İLE KODLAMAYA GİRİŞ

GİRİŞ

HTML sayfalarının içeriği, sayfayı ziyaret eden kişinin talepleri, kullanıcının tarayıcı veya cihazı ile bağlantılı olarak değişik biçimlerde görüntülenebilir. Bunu saptamak üzere HTML içeriklerine müdahale edebilen programlama dilleri geliştirilmiştir. Bu diller sunucu tarafı ve istemci tarafı programlama dilleri olarak ikiye ayrılabilir. Sunucu tarafı diller HTML içeriklerini sunucu üzerinde düzenleyerek sayfa yüklenmeden önce sunucu üzerinde HTML sayfalarını hazırlar. Ardından bu kodlar kullanıcının web tarayıcısına (istemci) gönderilir. Tarayıcıya ulaşan (indirilen) kodlar, web sayfası olarak gösterilmek üzere son kez derlenir. Bu derlenme aşamasında HTML, Javascript ve CSS kodları çalıştırılarak kullanıcının ekranına web sayfası oluşturulur.

Javascript, istemci tarafı bir dildir. Tarayıcılar tarafından çalıştırılır. HTML etiketleri ile oluşturulmuş web sayfasının içeriğini veya görünümünü ekranda görüntülenmeden önce veya sonra değiştirebilir.

JAVASCRIPT NEREYE YAZILIR?

Javascript kodları HTML sayfaları içerisinde iki farklı yere yazılabilir.

Javascript Komutlarını Ayrı Dosyalar Şeklinde Yazmak

Bunlardan ilki HTML sayfası dışında ayrı bir Javascript kod dosyası oluşturup (örneğin kodlar.js) bu dosyayı HTML sayfası içerisinde çağırma işlemidir. Bu yöntemde çağırma genellikle HTML sayfasındaki <HEAD>...</HEAD> etiketleri arasında yapılır. Ancak bu zorunlu değildir, sayfadaki başka bir noktada da çağrı yapılabilir. Çağırma komutu HTML etiketlerinden biri olan <script>...</script> ile yapılır. Örneğin <script src="kodlar.js"></script> komutudur, içinde bulunduğu HTML sayfası ile aynı klasörde bulunan "kodlar.js" dosyasını HTML sayfasına dahil eder.

Bu yöntemin birçok avantajı vardır. Örneğin aynı Javascript kodları başka bir HTML sayfasında kullanılacak ise bu yöntemle tüm sayfalara dâhil edilmesi (çağırılması) daha doğrudur, çünkü Javascript kodlarında herhangi bir değişiklik yapıldığında bu değişiklik tüm sayfalara yansımış olur. Ayrıca HTML sayfası içerisinde farklı dillerin (HTML, Javascript ve CSS gibi) bir arada bulunmasından dolayı geliştiricinin kafası karışabilir, bunu engellemek için Javascript ve CSS gibi kodları ayrı sayfalarda tutup gerektiğinde tek satır ile (<script src=""kodlar.js"></script> gibi) çağırma tekniği kullanılmalıdır.

Javascript Komutlarının HTML Sayfasına Açıkça Yazılması

İkinci yöntem ise Javascript komutlarını HTML sayfaları içerisine açıkça yazmaktır. Bu yöntemde yine <script>...</script> etiketleri kullanılır. Bu defa bu etiketlerin arasına Javascript komutları açıkça yazılır. Örneğin <script>document.write("Merhaba dünya"); </script> satırı, HTML sayfasında nerede kullanılırsa oraya "Merhaba Dünya" ifadesini yazar.

İkinci yöntemin tek avantajı, sadece bir yere özgü küçük Javascript komutları yazmak gerektiğinde dış dosyalara gerek duymadan işlem yapabilmektir. Ayrıca eğer dış dosyayı bulamama riski varsa Javascript komutlarını HTML sayfası içerisine yazmak daha güvenlidir.

JAVASCRIPT YAZMA, ÇALIŞTIRMA VE TEST ETME

Javascript tıpkı HTML sayfaları gibi metin tabanlı bir editörde yazılabilir. Çoğu geliştirici HTML sayfalarını kodlamak üzere kullandığı web editörü ile Javascript kodlamaktadır. Öte taraftan herhangi bir metin editöründe (Not Defteri® gibi) bu dosyalar açılıp yazılabilir.

Javascript Kodlarının Çalıştırılması ve Hatalı Kodların Tarayıcıda İzlenmesi

Javascript kodlarını çalıştırmak için kodları dâhil ettiğiniz HTML sayfasını tarayıcınızda açmalısınız. Bu aşamada tavsiyemiz, popüler olan tüm tarayıcılarda sayfanızı açmanızdır. Bunun sebebi, Javascript'in çok nadir de olsa farklı tarayıcılarda farklı şekilde çalıştırılmasıdır. Hatta günümüzde cep telefon ve tablet kullanımının artmasıyla denemelerinize bu cihazları da eklemenizi tavsiye ederiz.

Javascript komutları ile kullanıcının cihazında farklı kaynaklara erişilebilir. Bu durum kötü amaçlı yazım geliştiriciler tarafından kullanılabilir. Bu nedenle günümüz tarayıcıları kullanıcılara Javascript kodlarını çalıştırmayı iptal eden ayarlar yapmasına izin vermektedir. Dolayısı ile Javascript kodlarınızın bu şekilde çalışmadığı durumlarda kullanıcıların uyarılması ve sizin siteniz için izin vermesi istenmelidir.

Yazılan Javascript komutlarında yazım (sözdizim) hataları veya mantık hataları olabilir. Çoğu tarayıcı yazım hatalarını algılar ve hatanı olduğu satırdan itibaren Javascript komutlarını çalıştırmayı durdurur. Bu durumda web sayfasının HTML komutları yine de çalışır. Çoğu kullanıcı bunun farkına varmaz, ancak sayfadaki beklenmeyen içerik veya hareketleri fark edebilir. Geliştirici olarak yapmamız gereken tarayıcılardaki "geliştirici seçeneklerini" aktif hâle getirip, Javascript hatalarını görmek ve editörümüze dönüp düzeltmeler yapmaktır.

Eğer kullandığınız editör Javascript diline destek veriyorsa, kod yazma aşamasında yanlış yazımları fark ederek uyarı verir. Tavsiyemiz bu özellikte bir editör kullanmanızdır.

Kodlarda yapılan mantık hataları ise web sayfasında herhangi bir hata uyarısına neden olmaz ve geliştirici seçenekleri ile de ortaya çıkmaz. Bu hatalar çıktının farklı olması ya da kodların bir süre sonra çalışmaması ile ortaya çıkar. Bu durumda geliştirici tekrar kodlarına dönüp düzeltmeler yapmalıdır. Mantık hatalarından korunmanın en iyi yolu yazılan kodları farklı girdiler veya farklı ortamlarda çalıştırarak olası aksilikleri önleyecek kodları ilave etmektedir. Bu aşamada tarayıcıların "geliştirici seçenekleri" yine kodların çalışmasını izlemek için (hata olmasa bile) kullanılabilir.

JAVASCRIPT İLE YAPILABİLECEK İŞLER

Javascript ile yapılabilecek işler çok fazladır. Bu kitapta bunlardan en çok tercih edilenleri anlatmaya çalışılmıştır. Detayları daha sonra anlatılacak olsa da geliştiricilere fikir vermek amacıyla aşağıdaki işlerin Javascript ile yapılabilecek işler olduğu bilinmelidir.

Tarayıcı Algılama

Tarayıcı algılama, web sitesi sahiplerinin farklı tarayıcı kullanan kullanıcılara hizmet sunmasında son derece gerekli bir işlemdir. Farklı tarayıcılara göre farklı içerik, CSS veya Javascript komutları gerekebilir. Bunu algılamak için yine Javascript komutları kullanılır.

Bilgi Girişlerinde Eksik ya da Hatalı Girişleri Kontrol Etme

Bilgi girişlerinde eksik ya da hatalı girişleri kontrol etme, bir sonraki adımda hata olabilecek durumları engellemek için yapılması gereken işlemlerdendir. Örneğin e-posta adresi girilmesi gereken bir kutunun boş bırakılması ya da e-posta adresi olmayan bir ifadenin girişmiş olması Javascript ile kontrol edilebilir ve tespit edildiğinde daha fazla işlem yapmadan kullanıcı uyarılabilir.

Kullanıcı Hareketleri veya Tercihlerine Göre Farklı İçerik Sunma

Kullanıcı hareketleri veya tercihlerine göre farklı içerik sunma, sayfalarımıza etkileşim katmanının bir yoludur. Örneğin kullanıcının bir resim ya da metin üzerine fare ile gelmesi sonucunda o resim ya da metnin daha büyük görünmesi gibi etkileşimler Javascript ile yapılabilir.

Sayfa İçeriğinde Belirli Sürelerde Güncelleme Yapma

Sayfa içeriğinde belirli sürelerde güncelleme yapma günümüzde çok fazla yapılmaktadır. Anlık değişimleri çok olduğu web sitelerinde, her 10 saniyede ya da her 10 dakikada bir içeriğin belirli bir kısmının (tamamı da olabilir) yenilenmesi gerekebilir. Belirli periyotlarda yapılacak bu güncelleme Javascript ve sonucu tarafı dillerin iş birliği ile yapılabilir.

Matematiksel Hesaplamalar Yapma

Matematiksel hesaplamaların tarayıcı ekranında yapılması, kullanıcılar için basit hesaplamalarda farklı yazılımlara gerek duymadan sonucu hesaplamasını sağlar. Javascript ile hesap makinesi gibi oluşturulmuş ekranlar üzerinden işlem yapılabilir.

JAVASCRIPT DİLİNİN GENEL ÖZELLİKLERİ

Genel sözdizim özellikleri açısından Javascript C/C++ diline benzer. Bu kapsamda isimlendirme, komut

yazım stili ve nesne tabanlı programlama desteği açısında bu dillere benzerlik gösterir.

İsimlendirme Kuralları

Javascript'te programcı tarafından yaratılacak bir deęişken, fonksiyon ya da nesnenin isminde ancak A-Z ve a-z aralıęındaki harfler (sadece İngilizce harfler), rakamlar ve alt tire () karakteri kullanılabilir, dięer karakterler kullanılmaz. Karakterlerde büyük ve küçük harf ayrımı vardır. İlk karakter rakam olamaz.

Parantezler

Javascript dilinde açılan parantezler (normal () ya da küme {}) mutlaka kapanmalıdır. Aksi takdirde yazım hatası olur. Bu parantez ve içersine yazılacak komutların hizalanmış şekilde yazılması anlaşılabilirliği artıracaktır.

Komut Satırları ve Noktalı Virgül

Yazılan komutların sonuna, komutun sona erdiğini belirtmek üzere noktalı virgül (;) konur. Aksi takdirde sözdizim (yazım) hatası olacaktır. Sonlarına noktalı virgül konularak komutlar ister aynı satırda yan yanda, ister alt alta ayrı satırlar şeklinde yazılabilir. Alt alta yazmak okunabilirliği artırır.

Nesne Tabanlı Programlama

Javascript nesne tabanlı programlamaya destek verir. Sınıf ve nesne yaratmak mümkündür. Benzer biçimde HTML içeriklerine ulaşarak önlerin özelliklerini deęiştirmede de nesne tabanlı programlama notasyonunu kullanır. Ancak bu ünite de sınıf yaratma konusu ele alınmayacaktır.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI Javascript Dili ve Komutları
ÜNİTE NO 10
YAZAR Dr. Öğr. Üyesi HALİL ERSOY

JAVASCRIPT DİLİ VE KOMUTLARI

GİRİŞ

Javascript, dil özellikleri bakımında C/C++ programlama dillerine benzer. Komutlar noktalı virgül (;) ile biter, bloklar küme parantezleri { } ile açılır ve kapanır. Eğer istenirse tüm komutlar aynı satıra yan yana yazılabilir. Ancak kodların anlaşılır olması için komutların alt alta, her satırda tek komut olacak şekilde yazılması ve satır başı girintilerinin uygulanması tavsiye edilir. Bu şekilde yazılırsa, yazım ve olası mantık hataları daha kolay tespit edilir. Zira Javascript ve onu çalıştıran tarayıcılar herhangi bir eksikliği ya da yazım hatasını affetmez ve çalışmayı durdurur. Programcıların kodlamada tercih edeceği editörler, Javascript diline özgü hatırlatıcı ve yazım kontrol edici özellikte olmalıdır. Bu sayede kodlama aşamasında birçok olası hata fark edilebilir. Bu ünite, Javascript'te değişkenler, operatörler ve koşul ifadelerini ele alacağız.

JAVASCRIPT VE HTML ETİKETLERİ İLİŞKİSİ

HTML Etiketlerine Javascript ile Erişmek

Javascript dili ile HTML sayfalarındaki tüm etiketlerin özellikleri ve/veya içerikleri değiştirilebilir. Bunun için Javascript komutları içerisinde ilgili etikete document.getElementById() komutu ile erişilebilir. Komuttaki parantezler içerisine yazılacak isim (id), etikete verilmiş "id" ismi olmalıdır.

```
<p id="yazi1">Bu bir paragraftır.</p>
<script>
document.getElementById("yazi1").innerText="Bu yazı javascript ile değiştirilmiştir.";
</script>
```

Etiketlere verilecek "id" isimleri her sayfada benzersiz olmalı; isim içerisinde boşluk, noktalama işareti olmamalı; büyük küçük harf ayrımına duyarlı olduğu unutulmamalıdır. Etikete Javascript içerisinde erişmek için komut çalışmadan önce etiketin yüklenmiş olması gerekir.

Basit Mesajlar Yazdırma

Aşağıdaki 3 basit komut ile farklı yerlere çıktı yazdırma işlemi yapılabilir.

- alert();
- document.write();
- console.log();

İlk komut olan alert() komutu, parantezleri içerisine yazdığımız açık ifadeleri ya da değişkenlerin değerini, tarayıcıda açılacak küçük bir pencere içerisinde yazar. Kullanıcı bu pencereyi kapatmadan web sayfasını kullanamaz.

İkinci komut olan document.write() komutu parantezleri içerisine yazılan açık ifadeleri ya da değişkenlerin değerlerini, içinde bulunduğu <script> etiketlerinin arasında olacak şekilde web sayfasına yazar.

Üçüncü komut olan console.log() komutu ise yine parantezleri içerisinde yer alan değerleri, tarayıcının geliştirici seçeneklerindeki konsol paneline yazar. Bu daha çok geliştiricilerin hata ayıklama ya da programı izleme amacıyla kullandıkları bir komuttur çünkü web sayfasına bir çıktı vermez.

Açıklama Satırları

Javascript komutlarının arasına, çalıştırılması istenmeyen yazılar yazabilmek için açıklama satırı ya da satırları eklenebilir. Eğer tek bir satır açıklama haline getirilecek ise satır başına çift bölü işareti (//) konulmalıdır. Eğer birden çok satır açıklama haline getirilecek ise ilk satırın başına /* sembolü, son satırın da sonuna */ sembolü konulmalıdır.

```
// Bu satır bir açıklama satırıdır, çalıştırılmaz
```

```
/* Bu yazılar  
açıklama satırlarıdır,  
çalıştırılmazlar */
```

DEĞİŞKENLER

Javascript'te değişkenler mutlaka önceden tanımlanmalıdır. Tanımlama için var kelimesi kullanılır. Aşağıdaki 3 satırda tanımlanmış tüm değişkenler geçerlidir. Son satırda var kelimesi kullanılmamış olsa bir, ilk kez kullanılan ve ilk değeri verilen ortalama değişkeni geçerli olarak tanımlanmış olur.

```
var x, y, kareninAlanı;  
var a=30, b=3.1417, c="Aralık";  
ortalama=0.0;
```

Değişkenlerin isimlerinde a-z ve A-Z arasındaki karakterler (Türkçe karakterler hariç), rakamlar, alt tire (_) ve dolar (\$) sembolleri kullanılabilir. Değişken isimlerinde bir önceki ünite de belirtilen isimlendirme kuralları geçerlidir.

Değişken ve Veri Türleri

Değişkenlere tanılandıkları yerde ilk değer vermek tavsiye edilir. Bu sayede hem türü belirlenmiş olur hem de beklenmedik sonuçlar ortaya çıkmaz. Değişkenleri türü ilk alacakları değerin türü ile belirlenir. En sık kullanılan veri türleri sayı (number) ve metin (string) türüdür. Bunlardan başka belirsiz (undefined), nesne (object) ve ikilik (boolean) veri türleri mevcuttur.

Değişkenlerin veya açık değerlerin türlerini kod içerisinde anlamak için typeof() komutu kullanılır. typeof() komutu, parantez içerisine yazılan açık değerin ya da değişkenin türünü söyle (number, string, boolean, undefined, object).

Değişken Türlerinin Önemi

Değişken türlerinden uygun olanı seçmemiz gerekir. Bunun iki sebebi vardır:

- Bazı veri türleri hafızada (kullanıcının bilgisayarının RAM belleğinde) diğerlerine göre daha fazla yer işgal eder. Gereksiz yere büyük veri türleri seçmek, kullanıcının bilgisayarını gereksiz yere yoracaktır.
- Verinin türüne göre, veri üzerinde yapılacak işlemler farklılık gösterir, uygun olmayan türlerde yapılan işlemler yanlış sonuçlara neden olur.

Undefined ve Null Veri Değerleri

Değişkenler henüz ilk değerini almadıklarında tür ve değer olarak undefined şeklinde tanımlanır. Herhangi bir değişken açık biçimde undefined değerine atanarak hem içerisindeki değer silinmiş olur, hem de türü undefined olarak sıfırlanmış olur. Bu değişkene daha sonra farklı türde bir değer verilerek türü değiştirilebilir.

Null değeri ise nesnelerin (object türündeki değişkenlerin) değerini yok etmek için kullanılır. Null değerini alan nesnenin içindeki değer silinirken, türü yine object olarak kalır. Eğer nesnenin hem değerini hem de türünü sıfırlamak istenirse undefined değerine atanmalıdır.

OPERATÖRLER

Değişkenlerin değerlerini değiştirmeye veya değerlerini kontrol etmeye yarayan komutlara operatör denir. Operatörler genellikle sembollerden oluşur. Yaptıkları işler bakımından operatörler 3 sınıfa ayrılabilir:

- Aritmetik operatörleri,
- Karşılaştırma operatörleri,

- Mantık operatörleri.

Aritmetik Operatörleri

Sayısal değişkenler ve değerler üzerinde yapılabilecek işlemlerini gerçekleştiren operatörler aritmetik operatörleridir. Operatörleri ve isimlerini aşağıda görebilirsiniz.

- + Toplama
- - Çıkarma
- * Çarpma
- / Bölme
- % Bölümünden kalan (mod)
- ++ Sayıyı bir arttırmak
- -- Sayıyı bir azaltmak

Aritmetik operatörlerinden artırma (++) ve azaltma (--) haricindekiler 2 değer ya da değişken ile çalışır. Örneğin `a = b + 4;` komutu, `b` değişkeninin değeri ile 4'ü toplayıp sonucu `a` değişkenine aktarır. Ancak artırma (++) ve azaltma (--) operatörleri tek değişken ile çalışır ve kullanım şekline göre artırma ve ya eksiltme işlemini, atamadan önce ya da sonra yapar.

`a++;` // a'yı 1 arttırır

`++a;` // a'yı 1 arttırır

`b=a++;` // Önce b'ye a'nın değerini atar, ardından a'yı 1 arttırır

`b=++a;` // Önce a'yı 1 arttırır, ardından b'ye a'nın değerini atar

Bazı aritmetik işlemler, değişkenin kendisini kullanarak işlem yapıyorsa, bu işlemleri aşağıdaki tablodaki gibi kısaltarak kullanmak mümkündür.

- += Kendisiyle bir değeri toplama (`a+=4;` ve `a=a+4;` aynı şekilde çalışır)
- -= Kendisinden bir değeri çıkarma (`a-=3;` ve `a=a-3;` aynı şekilde çalışır)
- *= Kendisiyle bir değeri çarpma (`a*=2;` ve `a=a*2;` aynı şekilde çalışır)
- /= Kendisini bir değere bölme (`a/=3;` ve `a=a/3;` aynı şekilde çalışır)
- %= Kendisinin bir değere göre modunu bulma (`a%=10;` ve `a=a%10;` aynı şekilde çalışır)

Karşılaştırma Operatörleri

Değişken ya da değerlerin birbiri ile kıyaslanması için kullanılan operatörlere karşılaştırma operatörleri denir. Karşılaştırma operatörlerinin sonucunda `true` (doğru) ya da `false` (yanlış) değerleri üretilir. Bu değerler metin ya da sayı değildir, türü boolean'dır. Javascript'teki karşılaştırma operatörlerini ve anlamlarını aşağıda görebilirsiniz.

- == Değerleri birbirine eşitse
- != Değerleri eşit değilse
- === Değerleri ve türleri eşitse
- !== Değerleri ve türleri eşit değilse
- > Değeri büyükse
- < Değeri küçükse
- >= Değeri büyük veya eşitse
- <= Değeri küçük veya eşitse

Mantık Operatörleri

Birden fazla karşılaştırma işleminin birlikte değerlendirilmesi ve tek bir doğru (`true`) ya da yanlış (`false`) üretilmesi için mantık operatörleri kullanılır. Bu operatörler aşağıda verilmiştir.

- && Ve
- || Veya
- ! Değil

Ve (&&) operatörü, her iki yanında `true` ifadesi varsa, sonuç olarak `true`, bunun dışında tüm durumlarda `false` değerini üretir.

Veya (||) operatörü, her iki yanındaki ifadelerden en az biri `true` olduğunda sonuç olarak `true` değerini üretir. Eğer her iki yanında da `false` değeri varsa, o zaman sonuç olarak `false` değerini üretir.

Değil (!) operatörü, sağ rafındaki boolean türünden değeri tersine (`true` ise `false`, `false` ile `true`) çevirir.

Operatörlerde İşlem Önceliği

Yukarıdaki operatörler kendi başlarına iki ya da tek değişken ile işlem yaparlar. Buna karşın bazı işlemler birden fazla operatörün sırayla kullanılmasını gerektirebilir. Bu durumda farklı operatörlerin yan yana yazılarak sonucun üretilmesi gerekebilir. Böyle durumlarda yan yana yazılmış operatörlerin

hangisinin öncelikle çalışacağı bilinmelidir. Operatörler aynı satırsa yazılmış ilse aşağıdaki sıralama üstte olanlar altta olanlara göre daha önce çalışır.

- ++, --, -, ! (Sağdan sola)
- *, /, % (Soldan sağa)
- +, - (Soldan sağa)
- <, <=, >, >= (Soldan sağa)
- ==, !=, ===, !== (Soldan sağa)
- && (Soldan sağa)
- || (Soldan sağa)
- = (Sağdan sola)
- +=, -=, *=, /=, %= (Sağdan sola)

Eğer aynı satırda önceliği aynı olan bir den fazla operatör varsa, o zaman işlem yönüne göre (yukarıda belirtilen) işlem yapılır.

Eğer işlem önceliği bir başka operatöre verilmek istenirse, o operatör parantez () içine alınarak yazılır.

KOŞUL KOMUTLARI VE KOŞUL OPERATÖRÜ

Javascript'te iki adet koşul komutu vardır:

- if
- switch

Bu iki komuta ek olarak bir de koşul operatörü vardır:

- ?:

If Komutu

Bu komut, kendisi içerisinde yazılan kıyaslama işleminin sonucunda true değeri üretilirse devamındaki komutları çalıştırır. Sonuç false ise devamındaki komutları çalıştırmaz. Üç farklı şekilde kullanılabilir:

if(koşul)

```
    komutlar; if(koşul)
    komutlar1;
```

else

```
    komutlar2; if(koşul1)
    komutlar2;
```

else if(koşul2)

```
    komutlar2;
```

else if(koşul3)

```
    komutlar3;
```

else

```
    komutlar4;
```

Yukarıdaki tabloda belirtilen “komutlar” ifadesi, eğer birden fazla komut çalıştırılmak isteniyorsa, küme parantezi içerisine alınmış komut blokları ile değiştirilmelidir.

Switch Komutu

Komut değişkenin değerine göre çalıştırılacak komutları case ifadesi ile belirtir. Değeri değişkenin değeriyle eşleşen case komutları çalışır.

```
switch(değişken) {
```

```
    case değer1:
```

```
        komutlar;
```

```
        break;
```

```
    case değer2:
```

```
        komutlar2;
```

```
        break;
```

```
    .
```

```
    .
```

```
    .
```

```
    default:
```

```
        komutlar3;
```



```
    break;  
}
```

Switch komutundaki break komutu, kendisinden sonraki case içerisindeki komutların çalışmaması için eklenmelidir. Default seçeneği ise, isteğe bağlı olarak yazılır. Herhangi bir case değeri ile eşleşme olmazsa, default komutları çalışır

Koşul Operatörü (?:)

Bir koşula bağlı olarak, iki farklı değerden birini seçen operatör, koşul operatörüdür. Kullanımı aşağıdaki gibidir.

değişken = koşul ? değer1 : değer2;

Öncelikle koşulun sonucuna bakılır, sonuç true ise değer1 olarak ifade edilen değer değişken'e aktarılır. Eğer sonuç false ise değer2 olarak ifade edilen değer değişken'e aktarılır.



JAVASCRIPTTE İLERİ KONULAR

GİRİŞ

Bu ünite de Javascript dili ile yapılabilecek ileri düzey işlemler için gerekli olacak diğer konular ele alınmıştır. Bunlardan ilki metin (string) türü üzerinde yapılabilecek işlemlerdir. Bu işlemleri Javascript'in hazır fonksiyonları ile yapabiliriz. İkinci konu olarak döngüler ele alınmış, for, while ve do while döngülerle anlatılmıştır. Üçüncü konu olarak diziler anlatılmıştır. Dizilerin etkili kullanılabilmesi için döngülerle birlikte kullanılması gösterilmiştir. Devamında programlamanın temel unsurlarından biri olan fonksiyonlar konusu ele alınmıştır. Farklı türde tanımlama ve çağırılan fonksiyonlar anlatılmıştır. Beşinci konu olarak olalar (event) ele alınmıştır. Kullanıcının veya tarayıcının belirli durumlarına cevap verebilmek üzere olaylarda faydalanmak gerekir. Altıncı kısımda DOM (Document Object Model) model detaylıca anlatılmıştır. Bu model tarayıcı tarafında yaratılan ve Javascript benzeri betik dillerin yardımıyla içeriğin değiştirilebilmesini sağlayan standartları kapsar. Son olarak DOM içerisindeki window nesnesinin tarayıcı özellikleri, tarayıcı geçmişi ve açılı pencereler gibi metotları anlatılmıştır.

METİN TÜRÜ VE İŞLEMLERİ

Metin (string) türündeki değişkenler, tuttukları değerleri karakter dizisi olarak tutar. Her bir karaktere, dizinin bir elemanı olarak erişilebilir, ancak bu yolla değiştirmezler. Metin türündeki değerleri değişkenlere aktarmak ya da açık biçimde bazı fonksiyonlara parametre olarak göndermek için çift tırnak ya da tek tırnak içerisinde alınmalıdır.

Metinler üzerinden yapılabilecek işlemler aşağıdaki gibi hazır fonksiyonlar ile yapılabilir:

- `Metin.length` □ karakter sayısı veya metin uzunluğu
- `Metin.toUpperCase()` □ Metindeki harflerin büyük harfe dönüşmüş halini geri döndürür
- `Metin.toLowerCase()` □ Metindeki harflerin küçük harfe dönüşmüş halini geri döndürür
- `Metin.indexOf("aranan")` □ Metnin içindeki "aranan" ifadesini arar ve bulunduğu ilk yerin indis numarasını verir.
- `Metin.lastIndexOf("aranan")` □ Metnin içindeki "aranan" ifadesini arar ve bulunduğu son yerin indis numarasını verir.

DÖNGÜLER

Yapılacak işlemlerin tekrarlanması gerektiğinde aynı ya da benzer kodları tekrar yazmak yerine, işlemler döngü içerisinde yazılır. Döngüler belirli koşullar yerine getirildiği sürece kapsadıkları komutların tekrar çalışmasını sağlayan yapılardır. Eğer yanlış koşul yazılırsa döngünün hiç çalışmama ya da çalışmaya başladıktan sonra hiç sonlanmama (sonsuz göndü) tehlikesi vardır. Toplam 3 adet döngü komutu mevcuttur: For, while ve do while.

For Döngüsü

For döngüsünde genel kalıp aşağıdaki gibidir:

```
for(sayac=ilk değer ; sayac < sınır değeri; sayac++){  
    döngü içerisinde yapılacak işlemler  
}
```

While Döngüsü

While döngüsünde yine koşul döngünün başındadır. Diğer işlemler (gerekliyse ilk değer verme ve koşulun sağlanması için artış ya da azalış) döngü öncesinde veya döngü içerisinde yapılmalıdır. Genel yapı şu şekildedir:

```
sayac=ilk değer;  
while(sayac < sınır değeri){
```

```
döngü içerisinde yapılacak işlemler
sayac++;
}
```

Do While Döngüsü

Do while döngüsü while'a çok benzer, tek fark koşulun döngünün sonunda yer almasıdır. Bu durumda döngü bloğu bir kez koşulsuz çalıştırılır, devamında koşula göre tekrar çalıştırılır. Genel yapı şu şekildedir:

```
sayac=ilk değer;
do{
    döngü içerisinde yapılacak işlemler
    sayac++;
} while(sayac < sınır değeri);
```

Döngüler koşul değeri tekrar etmeyi gerektirse bile, gerektiğinde durmasını sağlayan komut break komutudur. Yukarıdaki üç döngü türünde break komutu döngüyü durdurur ve program akışını döngü sonrasına taşır.

Döngünün içerisinde program akışını kesip yine döngü başına yönlendiren komut ise continue komutudur.

DİZİLER

Diziler birden fazla değeri tek bir isim ile tutmayı sağlayan değişken benzeri yapılardır. Farklı türdeki değerler aynı dizide tutulabilir. Diziler tanımlanırken ilk değerleri dolu veya içi boş olarak tanımlanabilir.

```
var dizi1=[2,4,6,8,10]; □ 5 elemanlı dizi
var dizi2=new Array(5,10,15,20); □ 4 elemanlı dizi
var dizi3=new Array(5); □ 5 elemanlı, elemanları sıfır olan dizi
```

Dizi Tanımlama

New Array() komutu tek bir tam sayı ile çağırılırsa bu sayıyı eleman sayısı olarak algılar. Ancak iki ya da daha fazla sayı veya farklı türden parametreler ile çağırılırsa bunları dizinin ilk elemanları olarak algılar.

Dizilerin Elemanlarına Erişim ve İndis Numaraları

Dizilere atanan değerlere o değerlerin sırasına göre indis numarası ile teker teker erişilebilir ve değiştirilebilir. İndis numarası her zaman 0'dan başlar.

Dizinin Eleman Sayısı

Dizilerin eleman sayısına "dizinin uzunluğu" da denilebilir ve diziAdi.length ifadesi ile tamsayı olarak elde edilebilir.

Dizilerin Elemanlarına Döngüler ile Erişim

Dizilerin elemanlarının her biri için yapılacak işlemler genellikle aynı olduğu için bu işlemlerin döngüler içerisinde yapılması tercih edilir. Dizilerin eleman sayılarının belirli olduğu için (length özelliği ile) bu işlemlerde en çok for döngüsü tercih edilir.

Dizi Elemanları İçinde Arama Yapma

Dizilerin elemanları arasında arama yapmak için indexOf ya da lastIndexOf fonksiyonları kullanılır.

FONKSİYONLAR

Birden fazla komutu gruplayarak tek bir isim ile çağırabildiğimiz yapılara fonksiyon denir. Fonksiyonlar ancak tanımlandıktan sonra çağırılabilirler. Tanımlama kısmı ve çağırma kısmı genellikle aşağıdaki kalıba uygun biçimde yapılır.

// Tanımlama

```
function fonksiyonAdi (varsaParametre1, varsaParametre2, ... )
{
    Fonksiyon içinde çalışması istenen komutlar....
    return varsaGeriDönüşDeğeri;
}
```

Çağırma 2 şekilde olabilir: Geri dönüş değeri varsa:

```
var değişken= fonksiyonAdi(argüman1, argüman2...)
```

Geri dönüş değeri yoksa:

```
fonksiyonAdi(argüman1, argüman2);
```

Parametreler ve Argümanlar

Parametreler fonksiyonun, eğer istenirse, çağırıldığı noktadan değişkenler ya da değerler almasını sağlar. Tanımlama başlığında parantez içerisinde bir ya da birden çok parametre alabilir, eğer parametre istenmezse o zaman parantezlerin için boş bırakılır.

Tanımlama kısmında eğer parametre eklenirse, o zaman o fonksiyon çağırılma kısmında da aynı sayıda argüman ile çağırılmalıdır. Argümanlar değişken ya da açık bir değer olabilir. Parametrelerin değeri fonksiyon içerisinde değişebilir, ancak bu değişiklik temsil ettikleri argümanlara her zaman yansımaz.

Return Komutu

Eğer fonksiyon içerisindeki işlemlerin sonucunda, fonksiyonun çağırıldığı noktaya bir değer göndermek istenirse, o zaman fonksiyonun uygun yerine return kelimesi ve yanına uygun bir değer ya da değişken yazılır.

Fonksiyonların Çalışma Zamanı

Fonksiyonlar 3 farklı zamanda çalışabilir: Fonksiyonun adını varsa parametreleri ile birlikte yazıp bir komut gibi çağırma yöntemi ilk çalışma şeklidir. İkinci yöntemde HTML elemanlarının olaylarına yönelik yazılan fonksiyonların olay gerçekleştiğinde çalışmasıdır. Son yöntemde ise fonksiyon tanımlandığı yerde ve o anda çalışır.

Fonksiyonların JS Dosyalarına Yazılması

Fonksiyonlar JS uzantılı Javascript dosyaları içerisine yazılıp, HTML sayfalarına <script> etiketinin “src” özelliği ile eklenebilirler. JS dosyası içerisindeki fonksiyonlar bu sayede farklı HTML dosyalarında çalışabilir.

OLAYLAR

HTML etiketleri sayfaya yüklenme anında ya da yüklendikten sonra kullanıcının hareketlerini algılayabilirler. Algılayabildikleri bu durumlara olay (event) denir. Algılanan bir olay sonucunda istenirse Javascript komutları ya da fonksiyonları çalıştırılabilir. Bunun için olay adının ve çalışması istenen Javascript komut ya da fonksiyonlarının ilgili etiket içerisine yazılması gerekir. Örneğin;

```
<h1 onclick="alert('başlığa tıkladınız');">Javascript Dersleri</h1>
```

ifadesinde, onclick kullanıcının H1 etiketinin faresi ile tıklaması olayını temsil eder. Devamındaki komutlar da tıklama olayı gerçekleştiikten sonra çalıştırılır. Buradaki komut yerine fonksiyon çağırma işlemi de yapılabilir.

En Sık Kullanılan Olaylar

HTML etiketlerinin hepsinin kendisine has algılayabildikleri olay listesi vardır. Bu çok uzun bir listedir. Bunun yerine sıklıkla kullanılan aşağıdaki olayları bilmek çoğu durumda geliştiricilere yeterli olacaktır:

- onclick □ HTML elemanına kullanıcının fare ile tıklanması
- onload □ <body>, <image> veya <frame> gibi elemanların yüklenmesinin tamamlanması
- onMouseOver □ Fare işaretçisinin elemanın üzerine geldiğinde
- onMouseOut □ Fare işaretçisinin elemanın üzerinden çekilmesi sırasında

DOM MODELİ

HTML sayfaları yüklendiğinde, tarayıcılar sayfadaki tüm elemanları (HTML etiketlerini) document ismindeki bir nesnenin alt nesneleri olarak belirler. Bu hiyerarşi sayesinde HTML sayfasındaki tüm etiketlerin içeriği, özellikleri, CSS stilleri ve alt etiketleri değiştirilebilir. Bu hiyerarşik yapıya Document Object Model (DOM) denir.

Javascript ile DOM kullanılarak HTML sayfalarına yeni etiketler eklenebilir veya var olanları çıkarabilir, etiketlerin özelliklerini değiştirebilir, CSS stiller veya sınıfları değiştirebilir. Daha önemlisi HTML etiketlerinin kullanıcı hareketlerine yönelik değişmesi için olaylar eklenebilir.

DOM ile Elemanlara Erişim

Tarayıcı tarafından yaratılan ilk nesne document nesnesidir. HTML sayfasındaki diğer etiketler (elemanlar) bu nesnenin alt elemanları olarak hafızada tutulur. Bu hiyerarşide herhangi bir etikete erişmek istenirse öncelikle o etiketin “id” özelliğine bir isim verilmeli, sonrasında document.getElementById() metodu ile o elemana erişim sağlanabilir. Bunun dışında birden çok eleman erişmek için aşağıdaki metotlar kullanılabilir:

- document.getElementsByName(“nameBilgisi”)
- document.getElementsByTagName(“EtiketAdi”)
- document.getElementsByClassName(“classBilgisi”)

Yukarıdaki metotlar bilgisi verilen elemanların tümünü geri döndürdükleri için genellikle bir diziye değer aktaracak şekilde kullanılırlar.

InnerHTML ve InnerText Özellikleri

Eğer bir HTML elemanının açılış ve kapanış etiketlerinin arasına yeni HTML etiketleri eklemek isterseniz ya da var olan HTML etiketlerini değiştirmek isterseniz, innerHTML özelliğini kullanınız. Öte taraftan eğer bir HTML elemanının açılış ve kapanış etiketleri arasına metin eklemek ya da var olan

metni değiřtirmek isterseniz, innerText özelliđini kullanınız.

DOM Elemanlarının Özellik ve Stillerini Deđiřtirmek

HTML etiketlerinin özellik (attribute) ve stil öğelerini (style) Javascript ile ařađıdaki gibi deđiřtirebilirsiniz.
document.getElementById("resim").style.border="1px solid black";

document.getElementById("resim").className="kucukResimSinifi";

WINDOW NESNESİ

Window nesnesi, kullanıcının tarayıcısının özelliklerinin toplandıđı ve Javascript ile programlanabilir alt özellik ve metotlara sahip nesnedir. Window nesnesinin alt elemanlarına ulařmak için window kelimesi yazılmasa da olur.

Tarayıcı Ebatları

Kullanıcıların tarayıcıları farklı ekranlarda ve farklı en ve boy deđerlerinde olabilir. Bu deđerleri piksel cinsinden öğrenmek için ařađıdaki nesne özellikleri kullanılabilir.

- window.innerWidth □ Tarayıcı penceresinin geniřliđi
- window.innerHeight □ Tarayıcı penceresinin yüksekliđi
- window.screen.width □ Ekran (monitör) geniřliđi
- window.screen.height □ Ekran (monitör) yüksekliđi

Tarayıcı Adresi

Window.location nesnesi, ziyaret edilen web sitesinin ve sayfasının tam adresini, sayfanın adını, bađlantı protokolünü verir. İstenirse başka bir web sayfası adresi window.location.assign() metoduna parametre olarak verilerek yeni sayfanın yüklenmesi sađlanır.

Tarayıcı Geçmiři Nesnesi

Tarayıcılar ziyaret edilen tüm sayfaları “geçmiř” adı altında bir liste řeklinde saklar. Bu listeye Javascript ile ulaşabilirsiniz. Örneđin kullanıcınızın herhangi bir işlemi yanlış yaptıđında bir önceki sayfaya gitmesini sađlayabilirsiniz. Bunun için window.history nesnesinin back() ve forward() metotları kullanılır.

Tarayıcı Adı ve Sürümünü Belirleme

Window.navigator nesnesi, tarayıcı hakkında bilgi veren bir nesnedir. Özellikle window.navigator.appVersion özelliđin tarayıcı hakkında metin türünde detaylı bilgi verir. Bu bilginin filtrelenerek veya parçalara ayrılarak tarayıcı hakkında kesin bilgiye ulaşılabilir.

Mesaj Kutuları

Kullanıcıya tarayıcının ierisinde aılarak kısa mesajlar vermeyi veya kullanıcıdan kısa bilgiler almayı sađlayan birkaç mesaj kutusu alternatifi vardır. Bunlar;

- window.alert() komutu
- window.confirm() komutu ve
- window.prompt() komutudur.

Aılır Pencere Ama

Bir çok web sitesinde var olan tarayıcının ierisinde yeni bir sayfa ya da sekme amak için window.open() metodu kullanılır. Bu metot alacađı parametreler ile yeni aılacak sayfayı ister var olan tarayıcı pencere ya ad sekmesinde, istenirse yeni tarayıcı penceresi veya sekmesinde aabilir.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI JQuery'e Giriş
ÜNİTE NO 12
YAZAR Öğr. Gör. Rafet Orçun MADRAN

Giriş

Web teknolojileri HTML standartlarının ilk kullanılmaya başlandığı 1993 yılından bu yana inanılmaz bir hızla gelişti. Özellikle farklı hizmet sektörlerinde yer alan birçok uygulamanın İnternet tabanlı olarak kullanılmaya başlaması hem tarayıcıların yeteneklerinin gelişmesine hem de kullanılan teknolojilerde çeşitliliğe yol açtı.

HTML tek başına zengin bir uygulama deneyimini son kullanıcıya yaşatabilecek bir altyapıya sahip değildi. Bu noktada devreye giren istemci-sunucu mimarisinin ürettiği çözümler de yeterli gelmemekteydi. Çözüm, Web sayfalarının yenilenmesine gerek kalmadan da veri alış-verişini ve etkileşimi mümkün kılacak teknolojilerin kullanılmasıydı. jQuery bu tür çözümler sunan JavaScript kütüphaneleri içerisinde en yaygın kullanılan çözüm oldu.

jQuery Nedir?

jQuery, hızlı, küçük ve zengin özelliklere sahip bir JavaScript kütüphanesidir. Çok sayıda tarayıcıda çalışan kullanımı kolay uygulama programlama arayüzü ile HTML belgesinde gezinme, olay işleme, animasyon gibi uygulamaları çok daha basit hale getirir. jQuery, açık kaynak kodlu bir projedir. Projenin tüm kodları MIT lisansı ile kullanıma sunulmuştur.

JavaScript Kütüphanesi

Programlama dillerinin standart yapılarına ek olarak kullanılabilecek özellikleri içinde barındıran kod bloklarını kütüphane olarak tanımlayabiliriz. Kütüphaneler belirli bir amaç çerçevesinde birçok tanımlamayı ve hazır fonksiyonu içlerinde bulunduran yapılar olarak kod yazma süreçlerini çok kolaylaştırmakta ve geliştiricilere hızlı uygulama geliştirme olanağı sunmaktadırlar. jQuery, özellikle sayfa içi etkileşim ve basit animasyonların gerçekleştirilebilmesini çok kolay hâle getiren bir JavaScript kütüphanesidir.

jQuery Nasıl Çalışır?

JavaScript kütüphanelerinin genel çalışma prensipleri jQuery için de geçerlidir. Aslında basit bir metin dosyasından ibaret olan kütüphane dosyasına HTML kodunun yazıldığı dosyadan bağlantı verilmesi yeterli olmaktadır. jQuery kütüphane dosyasının sürekli güncel kalması isteniyorsa kütüphanenin İnternet üzerindeki depodan kullanılması uygun olacaktır. Ancak eğer geliştirilecek uygulama İnternet'e bağlı olmayan bir platform için ise kütüphane dosyasının yerel bir adreste bulunması gerekir.

jQuery Kurulumu

jQuery ile ilgili kurulumda ihtiyaç duyacağımız tüm kaynak dosyalar <http://jquery.com/download/> adresinde yer almaktadır. Bu sayfada jQuery dosyalarının hem sıkıştırılmış hem de sıkıştırılmamış sürümleri yer alır. Sıkıştırılmamış dosya biçimi daha çok kod üzerinde geliştirme çalışması yapmak isteyenler için hazırlanmıştır; okunması, üzerinde değişiklik yapılması daha kolaydır. Sıkıştırılmış dosya biçiminin avantajı ise boyut olarak daha küçük olmasıdır. Bu şekilde HTML sayfalarının İnternet tarayıcıları tarafından yüklenmesi daha hızlı olur.

jQuery Sözdizimi

jQuery sözdizimi HTML öğelerinin (etiketlerinin) ve CSS tanımlamalarının seçilmesi ve bu öğelere birtakım eylemler atanması şeklinde gerçekleştirilir. jQuery söz dizimi içerisinde önemli noktalardan biri de "The Document Ready Event" adı verilen doküman hazır olayıdır. Bu olay genel kullanımda jQuery kodlarını kapsayıcı bir fonksiyondur. Kullanılan bu fonksiyon sayesinde dokümanın yüklenmesi henüz tamamlanmadan jQuery kodlarının çalışması engellenir ve olası bir hatadan kaçınılmış olur.

jQuery Seçicileri (Selectors)

jQuery seçicileri HTML öğelerinin seçimini ve bu öğeler ile ilgili değişiklikleri mümkün kılar. Seçiciler HTML öğelerini adlarına, kimliklerine (id), sınıflarına, tiplerine, niteliklerine, niteliklerinin sahip olduğu değerlere göre seçebilirler. jQuery'nin birçok farklı şekilde seçim işlemini gerçekleştirebilir. Bir HTML sayfası içinde yer alan tüm paragrafları değil, sadece belirlediğimiz paragrafların işleme alınmasını isteyebiliriz. Bu noktada bir HTML etiketi değil, etiketin içindeki özel bir sınıfı ya da seçim işlemi için kullanabiliriz.

jQuery Olayları

jQuery olayları, HTML sayfası içindeki etkileşimlere karşılık vermenizi mümkün kılar. Bu etkileşimler farenin bir HTML ögesi üzerinden geçmesi, bir form elemanın seçili hâle getirilmesi ya da HTML sayfası içerisindeki herhangi bir ögeye tıklanması şeklinde ortaya çıkabilir.

jQuery EFEKTLERİ

jQuery'nin kullanıcıları en çok etkileyen özellikleri HTML öğeleri ile yapılan efektlerdir. Bu efektler sayesinde sayfada yer alan öğeler gizlenebilir ya da gösterilebilir. Görsellere kaydırma efekti uygulanarak slayt gösterileri (günümüz Web sitelerinin neredeyse temel özelliklerinden biri hâline gelmiştir) düzenlenebilir. Web sayfalarında sıklıkla kullanılan jQuery efektleri aşağıda listelenmiştir:

- Gizle / Göster (Show / Hide): jQuery ile HTML elemanlarını `hide()` ve `show()` metotları ile gizleyebilir ya da görünür hâle getirebilirsiniz.
- Soldurma (Fade): jQuery ile HTML elemanlarını “fade” metodunu kullanarak soldurma efekti ile gizleyebilir ya da belirginleştirme efekti ile görünür hâle getirebilirsiniz.
- Kaydırma (Slide): jQuery ile HTML elemanlarını “slide” metodunu kullanarak kaydırabilirsiniz.
- Hareketlendirme (Animate): jQuery ile HTML elemanlarına kendi belirlediğiniz şekilde (önceden tanımlanmış efektlerden bağımsız olarak) farklı hareket ve efektler vererek animasyonlar oluşturabilirsiniz. Bu animasyonlar HTML elemanlarının sayfa üzerindeki konumlarını değiştirerek, renk ve boyutları üzerinde değişiklikler yaparak gerçekleştirilmektedir.
- Durdurma (Stop): jQuery'deki `stop()` fonksiyonu uygulama içerisindeki animasyonların henüz bitmeden önce durdurulması için kullanılır.

jQuery efektleri için kullanılan metotlarda “Toggle” adlı özel bir metot türü bulunmaktadır. Bu metot türü ile tek bir tetikleyici kullanarak (örneğin Web sayfasında yer alan tek bir buton yardımıyla) işlemin ilk tetiklemede (fare örneğinde tıklamada) gerçekleşip, ikinci tetiklemede ise (fare örneğinde yeniden tıklamada) işlemin geri alınması sağlanabilmektedir.



GİRİŞ

İnternet'e bağlı bir bilgisayardan çok farklı hizmetlere (e-devlet, e-finance, e-ticaret, vb.), kaynaklara erişilmesi ve bunun sağlamış olduğu avantajlar kullanıcıların bu sürece mobil ortamda da devam etme taleplerini beraberinde getirmiştir. Bu talepleri mobil platformlarda karşılayabilmek için farklı yazılım çözümleri hayata geçirilmiştir. Bu çözümler özel olarak mobil cihazlar için geliştirilmiş olan "Mobil Uygulamalar" ve "Mobil İnternet Tarayıcılar" ile sağlanır.

Mobil cihazlarıyla Web sitelerini ziyaret eden kullanıcıların iyi bir kullanıcı deneyimi yaşayabilmeleri için tercih edilen geliştirme ortamlarından biri JQuery Mobile'dır. JQuery Mobile, sadece mobil cihazlar için değil, dokunmatik ekrana sahip birçok farklı cihaz (kiosklar, araç navigasyon sistemleri vb.) için de uygulama geliştirme ortamı olarak kullanılabilir.

JQUERY MOBILE NEDİR?

JQuery Mobile, HTML5 temelli bir kullanıcı arayüz tasarım sistemidir. Özellikle sayfa içi etkileşim ve basit animasyonların gerçekleştirilebilmesini çok kolay hâle getirir. Bunlara ek olarak kullanıcı arayüzünü oluşturacak özel olarak mobil cihazlar için yapılandırılmış HTML elemanları da içerir.

JQuery Mobile Paketi Neler İçerir?

JQuery Mobile Paketi hem JavaScript kütüphanelerini hem de CSS adını verdiğimiz görsel arayüzü oluşturan tanımlamaları içerir. Paket içeriği kullanılacak ek özelliklere göre farklı kütüphaneler ve CSS tanımlama dosyaları içerebilir.

JQuery Mobile Nasıl Çalışır?

Çatı (framework) adını verdiğimiz farklı bileşenlerden oluşan Web mimarilerinin genel çalışma prensibi JQuery Mobile içinde geçerlidir. HTML5 tarzı bir çalışma sistemine sahip olan JQuery Mobile, JQuery gibi sadece JavaScript kütüphanelerinden oluşmaz, CSS tanımlamalarını içeren dosyaları da kullanır. JQuery Mobile çekirdek sistemini oluşturan tüm dosyalar yerel bir adresten (bilgisayarın sabit diskinden) ya da İnternet üzerindeki depolardan (CDN) kullanılabilir.

JQUERY MOBILE KURULUMU

JQuery Mobile ile ilgili tüm kurulum dosyaları ve ayrıntılı dokümantasyon

<https://jquerymobile.com/download/> adresinde yer almaktadır. Bu adresten indirilen JQuery Mobile paketinin içinde çekirdek sistem dosyalarına, Web sayfasının başlık (<head>) bölümünden referans verilerek bağlantı sağlanır. Bu bağlantı İnternet üzerindeki bir depo adres gösterilerek de sağlanabilir.

JQuery Mobile Sayfa Yapısı

Bir JQuery Mobile sitesi, çatının (framework) tüm özelliklerinden tam olarak yararlanabilmek için bir HTML5 doküman tipi tanımlaması ile başlamalıdır. Sayfanın <head> etiketleri içerisinde yer alan başlık bölümünde JQuery, JQuery Mobile ve CSS mobil temasına referans verilmesi zorunludur.

Sayfanın başlık bölümünde ek bazı üst veri tanımlamaları da yapılır. Bu tanımlamalarından biri de "Viewport" üst veri etiketidir. Bu etiket yardımıyla mobil İnternet tarayıcıları masaüstü bir siteyi değil, mobil cihazlar için geliştirilmiş bir sayfayı görüntülediğini anlar ve cihazın ekranının piksel genişliğine göre içeriği orantılayarak gösterir.

Web sayfasının başlık bölümündeki tanımlamaları, Web sayfasının gövde bölümündeki özel JQuery Mobile sayfa tanımlaması takip eder. Bu sayfa tanımlaması üst bilgi, içerik ve alt bilgi alanlarından oluşan bir sayfa şablonunun oluşturulmasını sağlar. Bu sayfa şablonu, geliştirilecek mobil Web sitesi için bir temel teşkil eder ve bu temelin üzerine tüm site inşa edilir.

Tek sayfa şablonuna ek olarak, içinde birden çok sayfanın barındırılabilceği çoklu sayfa şablonu da mobil Web sitesi geliştirilmesinde kullanılan yöntemler arasındadır. Bu şablon yapısında, tek bir HTML dokümanının içerisinde birden çok JQuery Mobile sayfa yapısı oluşturulur ve bu sayfalar arasında farklı efektler ile geçiş yapmak mümkündür.

JQuery Mobile Sayfa Bileşenleri

JQuery Mobile, sayfa içerisinde birçok farklı HTML elemanını ve bu elemanlar yardımıyla oluşturulmuş nesneleri kullanmamıza olanak sağlar. Bu nesneler özel olarak JQuery Mobile için yapılandırılmışlardır

ve bileşen (widget) olarak ifade edilirler.

Bileşenler, bir mobil uygulamada kullanılan arayüz elemanlarının sağladığı kullanıcı deneyimini bir mobil tarayıcıda deneyimleyebilmemize olanak sağlar. Bu bileşenler kimi zaman navigasyon ve ek bilgi sağlayabilecek hareketli paneller olarak karşımıza çıkar, kimi zaman ise Web sayfalarında verileri daha düzenli yayınlayabilmemiz için kullanabileceğimiz listeler ve tablolar şeklinde kullanıma sunulur.

jQuery Mobile'ın en önemli sayfa bileşenlerinden bir de formlardır. Formlar, web sayfasında kullanıcı etkileşimi sağlayabilmenin en önemli yoludur. Bu etkileşimi sağlayan form bileşenleri, standart HTML form elemanlarının (metin kutuları, onay kutuları, radyo düğmeleri vb.) jQuery Mobile'ın CSS tanımlamaları ile özel olarak biçimlendirilmiş halleridir. Standart HTML form elemanlarına ek olarak bir takım özel bileşenler de (ör: kaydırıcılar (slider), şalter görünümlü düğmeler (flip switch)) form bileşenleri altında kullanılabilir.

JQUERY MOBILE TEMALARI

jQuery Mobile içerisinde Web sitemizin genel yapısı ile görsel olarak uyum sağlayacak farklı temaları oluşturabilme ve bu temalar arasında geçiş yapabilme şansımız bulunmaktadır. Bu temaları oluştururken Web tabanlı bir arayüze sahip olan Theme Roller işimizi çok kolaylaştırır. Gerçek zamanlı olarak temalar üzerinde yaptığımız değişiklikleri Web tabanlı olarak görebildiğimiz bu tema oluşturucu aynı zamanda ihtiyacımız olan dosyaları da bize otomatik olarak paketler ve indirilmeye hazır bir hâlde sunar.

Theme Roller'da oluşturduğumuz tema paketlerinin içeriğini kendi proje klasörümüze kopyalamak yeni temayı kullanmaya başlamanız için yeterlidir.



DERS ADI İnternet Programcılığı I
ÜNİTE ADI Bootstrap ile Duyarlı Web Tasarımı
ÜNİTE NO 14
YAZAR Öğr. Gör. Rafet Orçun MADRAN

GİRİŞ

Mobil cihazların ilk akıllı sürümleri piyasada yer almaya başladığı zamanlarda Web sitelerinin masaüstü ve mobil sürümlerini iki ayrı site olarak tasarlamak sıkça kullanılan bir yöntemdi. Farklı ekran boyutlarına sahip çok sayıda cihazın piyasaya girmesi ile bu çözüm yetersiz kaldı.

Tek bir arayüz tasarımının birden çok cihaz ile uyumlu bir şekilde çalışabileceği bir sistem üzerinde durulmaya başlandı. Bu çalışmalar Duyarlı Web Tasarımı adı verilen bir kavramı ortaya çıkardı.

BOOTSTRAP NEDİR?

Duyarlı Web Tasarımı'nı zorlu bir sürece dönüştüren teknik altyapı gereksinimlerini karşılayabilecek bütünleşik sistemlere ihtiyaç duyuldu. Bu ihtiyacı HTML5 çatıları (framework) karşıladı. Bootstrap, mobil cihazlar için öncelikli, duyarlı web siteleri geliştirmeyi sağlayan, açık kaynak kodlu, ızgara sistemine sahip ve jQuery üzerine inşa edilmiş güçlü bileşenleri ile popüler bir HTML5 çatısıdır.

Bootstrap Paketi Neler İçerir?

Bootstrap paketinin içeriğini iki farklı dosya türü oluşturur; CSS tanımlamalarının yer aldığı *.css uzantılı dosyalar ve JavaScript kütüphanelerinin yer aldığı *.js uzantılı dosyalar.

Bootstrap Nasıl Çalışır?

Bootstrap'in kullanılacağı HTML dosyasından ilgili dosyalara (css ve js) referans verilerek bağlantı sağlanması çatının aktif hale gelmesi için yeterlidir. Referans verilme işlemi gerçekleştirildikten sonra JavaScript kütüphanesindeki fonksiyonları ve CSS tanımlamalarını kod içinden çağırarak istenilen yapı inşa edilebilir.

BOOTSTRAP KURULUMU

Bootstrap kurulumu iki farklı şekilde gerçekleştirilebilir; Bootstrap çekirdek dosyaları İnternet üzerinde yer alan depolardan kullanılabilir ya da çekirdek dosyalar bilgisayarın sabit diskine kaydedilerek bağlantı yerel olarak sağlanır.

Bootstrap kurulumunda dikkat edilmesi gereken noktalardan biri klasörlerin ve dosyaların konumları ve isimlendirilmeleridir. Web sitelerinin zamanla artan dosya ve klasör sayıları, ileriki süreçte projeyi güncellemesi ve bakımı zor bir hale getirebilir. Bu açıdan ana proje klasörü altında dosyaları özelliklerine göre alt klasörlere ayırmak iyi bir çözüm olmaktadır.

DUYARLI WEB SAYFASI TASARIMI

Bootstrap'in en önemli özelliklerinden biri duyarlı Web sayfaları inşa edebilmektir. Duyarlı bir Web sayfasının temel özelliği ekran boyutuna göre sayfa öğelerinin yeniden yerleşiminin sağlanabilmesidir. Buna ek olarak yazıtipi ile ilgili yeniden boyutlandırma, menü çubuğunun yeniden yapılandırılması vb. özellikler farklı cihazlar için uyumlu tasarımların hayata geçirilmesini sağlar.

Duyarlı Web Sayfası Şablonu

Bir Web sitesi tasarımında şablonlar ile çalışmak geliştirici açısından pratik ve hızlı üretim süreci anlamına gelir. Altyapısı iyi kurgulanmış bir Web sayfası şablonu, Web sitesinin inşa sürecini kolaylaştıracaktır.

Duyarlı Web sayfası şablonlarında standart HTML dosyalarındaki yapı kullanılır. Başlık bölümünde gerekli kütüphane ve tanımlama dosyalarına referanslar, gövde bölümünde ise kullanıcıya sunulacak içerik yer alır. Sayfa şablonu hazırlandıktan sonra İnternet tarayıcısı üzerinde test edilir, bu test sırasında tarayıcı penceresi boyutlandırılarak farklı ekran boyutları simüle edilebilir.

BOOTSTRAP İLE WEB SİTESİ İNŞASI

Şablonun test işlemi gerçekleştirildikten sonra, navigasyonu sağlayacak menü çubuğu da şablona eklenir. Menü çubuğunun da duyarlı bir yapıda olması gerekir. Bootstrap, menü çubuğu ile ilgili de çözümler sunmaktadır. Web sayfasının genel yapısı tamamlandıktan sonra, istenilen sayıda kopyalar çıkartılarak Web sitesini oluşturacak tüm sayfalar oluşturulur ve genel navigasyon içerisinde tanımlanırlar.

Web sayfalarının oluşturulduğu şablon, içerik bölümünün farklı şekilde yapılandırılması için de uygundur. Farklı sütun sayılarından oluşan bölmeler, görsel öğelerin farklı boyutta ve konumda

kullanılmaları mümkündür. Genel duyarlılığı sağlayacak yapı bozulmadıkça içerik alanında istenilen değişiklikler yapılabilir.

BOOTSTRAP WEB SİTESİ ÖRNEKLERİ

Bootstrap yaygın kullanımından dolayı İnternet üzerinde çok fazla örnek bulabileceğiniz bir çatıdır. Bu örnekler haricinde beğendiğiniz herhangi bir Web sitesinin kodlarına bakarak (ki bu kodları standart bir İnternet tarayıcıda görüntüleyebilmeniz için hiçbir engel yok), bu sitenin altyapısını hangi çatı ya da sistemlerin oluşturduğunu da anlayabilirsiniz.

Bootstrap'in kendi Web sitesi içerisinde bulunan örnekler bölümü, yeni bir Web sitesinin tasarımına başlamak için ideal bir nokta oluşturmaktadır. Örnekler bölümünden seçeceğiniz herhangi bir Web sitesini İnternet tarayıcınız yardımıyla sabit diskinize kaydedebilir ve üzerinde değişiklikler yaparak kendi Web sitenizi bu altyapı üzerinde inşa edebilirsiniz.