

Form Tasarımı ve Veri İletimi

Form Yapısı

- 1 Formların kullanım amaçlarını ifade edebilme
- 2 Genel form yapısını tanımlayabilme
- 3 Form özelliklerinin kullanım amaçlarını ifade edebilme

Form Elemanları

- 4 Form elemanlarını ve HTML5 ile gelen form elemanlarını listeleyebilme
- 5 HTML5 kavramını açıklayabilme
- 6 Birden fazla form elemanı ile WEB formu tasarlayabilme
- 7 Form elemanlarının özelliklerini kullanabilme

Anahtar Sözcükler: • HTML • WEB • Form • Form Elemanları • Form Özellikleri • Veri İletimi

Username*

Your username

Password*

Remember me

LOGIN

OR

GİRİŞ

WEB, İnternet üzerinde en çok kullanılan servislerden biridir. WEB sayfaları aracılığıyla bilgiye erişimdeki ve iletişimdeki yenilikler hayatı neredeyse her alanda değiştirmektedir. Teknolojiyle birlikte yeni iş ortamları yaratılırken mevcut işlerin yapılmasında yeni şekiller ortaya çıkmaktadır. İnsanlar sosyal ağlara dâhil olarak farklı iletişim olanaklarına kavuşmakta, bilgisayarlar da kendi aralarında haberleşebilmektedir. Bunun yanında WEB'in tarihine bakıldığında WEB 1.0, WEB 2.0, WEB 3.0 ... gibi bir sıradan söz edilebilir. İlk WEB sayfalarında bilgi girişi yapılabilecek arayüz elemanları ve teknolojileri olmadığından, İnternet kullanıcıları kendilerine sunulan bilgiye sadece bakabilmekteydiler. Siteler arasında gezinerek vakit geçirmek anlamına gelen WEB'de sörf tanımı da WEB 1.0 ile ortaya çıkan bir kavramdı. WEB 2.0, İnternet kullanıcılarının WEB sayfalarına bilgi girebildikleri yeni teknolojilerin ortaya çıkmasıyla anılmaya başlandı. Bu durum sosyal ağlar, yeni alışveriş, eğitim ve akla gelebilecek diğer sitelerin ortaya çıkmasını sağladı. Bugünün WEB anlayışında bilinen birçok sitenin içeriğini aslında kullanıcıların kendileri oluşturmaktadır. Başka bir deyişle aslında kullanıcılar olmasa bu siteler basit, belki de boş birer WEB sayfasından başka bir şey olmayacaktır. Yüklenen videolar, gönderilen mesajlar ve bilgi akışları aslında İnternet kullanıcılarının kendi ürünleridir.

Teknik anlamda WEB sayfaları, sunucu ve istemci olarak anılan bilgisayarların birbirlerine veri akışı yapmalarıyla iş görmektedir. Şöyle ki; Bir kullanıcı, bilgisayarı başında bir WEB sayfasına erişmek istediğinde, kendi bilgisayarını istemci olarak, erişmek istediği WEB sitesinin bilgisayarına yani sunucuya İnternet üstünden erişim sağlar. Sunucu üzerinde barındırdığı WEB sitesini ziyaretçinin bilgisayarı olan istemciye gönderir. Böylelikle kullanıcı istediği WEB sayfasına girmiş olur. Bu klasik akış WEB 1.0'ın çalışma biçimini de ifade etmektedir. WEB sayfalarının temelini oluşturan HTML (HyperText Markup Language) üzerinde yapılan geliştirmeler ve iyileştirmeler ve yeni WEB teknolojileri bu süreci bir adım daha ileriye götürmüştür. WEB 2.0'da sadece WEB sayfaları değil, kullanıcıya ait bilgiler de bilgisayarlar arasında dolaşmaktadır. WEB sayfaları üzerinde daha fazla etkileşim olanağı, HTML formları sayesinde gerçekleşmiştir. Yukarıda sözü geçen kullanıcıların WEB içeriğini kendilerinin oluşturmaları kavramı bu formlar sayesinde gerçekleşmiştir. Örneğin sosyal ağda kişisel sayfaya resim ekleme, alışveriş sitesinde adres girme, ya da bir haber sitesindeki habere yorum yapma formlar aracılığıyla gerçekleşir.

Kitabın bu bölümünde formlara, form elemanlarına, formların ve form elemanlarının özelliklerine yer verilmiştir. Konuyu en iyi şekilde anlatabilmek için kodlar ve bu kodların tarayıcıdaki görüntüleri alt alta konarak resimlendirilmiştir.

FORM YAPISI

Formlar genelde kullanıcıların veri girdiği ya da seçim yaptığı alanlar ile bu bilgileri sunucuya gönderen düğmelerden oluşan bir bütündür. Formlar içerisinde verilerin girildiği ya da seçimlerin yapıldığı araçlara form elemanları denir. Bir formun bir ya da birden fazla form elemanına ait verileri bir bütün olarak sunucuya gönderilebilir. Resim 5.1'de ad ve soyad bilgilerinin girilebildiği bir formun kodları yer almaktadır. Formları daha iyi anlayabilmek için bu örneğe göz atalım.

✓ Formlar genellikle kullanıcının bilgilerini ve seçimlerine ait verileri toplayan elemanlar ile bu verileri sunucuya gönderen düğmelerden oluşan bir bütün olarak düşünülebilir.

```
1 <form action="kayit.php" method="get" target="_blank">
2   Ad: <input type="text" name="isim">
3   Soyad: <input type="text" name="soyisim">
4   <input type="submit" value="Gönder">
5 </form>
```

Resim 5.1 Basit bir form örneğinin kodları

Resim 5.1'deki kayıt formunda isim, soyisim girilebilen iki metin kutusu ve bir gönder düğmesi olmak üzere üç adet form elemanını oluşturan kodlar yer almaktadır. Kodlar tarayıcıda açıldığında Resim 5.2'de ekran görüntüsü elde edilmektedir.

Ad: Soyad:

Resim 5.2 Basit bir form örneğinin tarayıcı görüntüsü

Kutucuklara girilen bilgiler Gönder düğmesiyle sunucuya gönderilmektedir. Bir siteye üye olmak istediğimiz zaman ad, soyad ve diğer bilgilerimizi bu tür formlara girerek siteye üye olmaktadır.

Resim 5.1'deki birinci satırda görüldüğü üzere `<form>` etiketi içinde action, method ve target gibi kodların bulunduğu görülmektedir. Bu kodlara özellik denilmektedir. Özellikler formlara ve form elemanlarına ek işlevler kazandırmak ya da sınırlamalar getirmek için kullanılırlar.

✓ Formlara ve form elemanlarına ek işlevler kazandırmak ve sınırlamalar getirmek için özellik olarak adlandırılan kodlar kullanılır.

Yukarıdaki form örneği formların genel yapısı hakkında bilgi vermektedir. Öyleyse bir formun genel yapısı Resim 5.3'te sunulduğu gibi özetlenebilir. Formlar `<form>` etiketiyle başlayıp klasik HTML kodlarında olduğu gibi `</form>` komutuyla sonlandırılır.

```
1 <form action="dosya yolu" method="get ya da post" ve diğer özellikler>
2 <form elemanları ve özellikleri>
3 </form>
```

Resim 5.3 Form yapısı



dikkat

Formlar `<form>` etiketiyle başlatıldıktan sonra kapatma etiketi olan `</form>` ile sonlandırılmalıdır.

Form Özellikleri

Resim 5.3'te görüldüğü gibi `<form>` etiketi içine Tablo 1'de yer alan özellikler eklenebilir. Bu özelliklerin alabilecekleri değerler, kullanım biçimleri ve tanımları tabloda yer almaktadır. Ayrıca bazı özelliklerin hemen altında ve sonraki bölümlerde görüleceği üzere HTML5 ibaresi yer almaktadır. HTML5, o özelliğin HTML5 ile yeni gelen bir özellik olduğunu göstermektedir. WEB sayfası tasarımının temelini oluşturan HTML, zaman içinde geliştirilerek yeni özellikler kazanmaktadır. Bu özellikler gerek WEB sayfası tasarımcılarına gerekse WEB kullanıcılarına kolaylıklar sağlamaktadır. Son olarak olarak HTML5 sürümü kullanılmaktadır.

Tablo 5.1 Form özellikleri

Özellik	Değer ve Kullanım	Tanımı
action	= "http://ornek.com/kayit.php" gibi	Form verilerinin sunucuda nereye gönderileceğini belirtir.
method	= "get" = "post"	Form verilerinin gönderilmesinde hangi HTTP yönteminin kullanılacağını belirtir.
enctype	= "application/x-www-form-urlencoded" = "multipart/form-data" = "text/plain"	Form verileri sunucuya gönderilirken nasıl kodlanacağını belirtir. Sadece method="post" ile kullanılabilir.
name	= "kayitformu" gibi	Forma bir isim verilmesini sağlar.
accept-charset	= "UTF-8" =	Form verilerinin hangi karakter kodlamasını kabul edeceğini belirtir.

novalidate HTML5	<i>novalidate</i>	Form verilerinin doğrulanmaması için kullanılır.
target	="_blank" (Yeni boş sayfa) ="_self" (Formun açıldığı sayfa) ="_parent" (İlk açılmış sayfa) ="_top" (En üstteki sayfa)	Form verileri sunucuya gönderildikten sonra cevabın hangi sayfada (Boş, açık olan, ana sayfa ya da en üstteki WEB sayfası) görüntüleneceğini belirtir.
autocomplete HTML5	="on" ="off"	Bu özellik form doldururken kullanıcıya önceden girilmiş bilgileri göstererek yardımcı olur.
accept	="audio/*" ses dosyaları ="video/*" görüntü dosyaları ="image/*" resim dosyaları =" .jpg, . doc" gibi (virgülle ayrılmış)	Sunucunun kabul ettiği dosya türlerini belirtir.

Kaynak: <https://www.w3.org/TR/html5/forms.html>, 2016

Tablo 5.1.'deki özellikler form yapısına uygun şekilde <form> etiketi içinde kullanılabilir. Örneğin Resim 5.4'te bir metin kutusu ve düğmeden oluşan bir formun kodları görülmektedir. Bu kodların tarayıcıdaki görüntüsü Resim 5.5'te yer almaktadır.

```

1 <form action="kayip.php" method="get" target="blank">
2   Adınız<br>
3   <input type="text" name="isim"><br>
4   <input type="submit" value="Gönder">
5 </form>

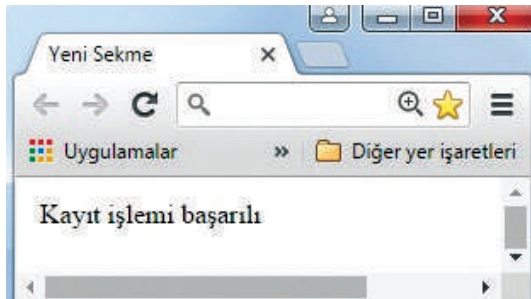
```

Resim 5.4 Formlara özellik eklemek

Adınız

Resim 5.5 Diğer form özelliklerini ekleme kodlarının tarayıcı görüntüsü

Metin kutusuna isim yazıldıktan sonra gönder düğmesine basıldığında, girilen değer isim değişkeniyle sunucuya gönderilmektedir. Sunucuda gerekli işlemler yapıldıktan sonra kayıt işleminin sonucu yeni bir pencerede "Kayıt işlemi başarılı" şeklinde gösterilmektedir. İşlemin tarayıcı görüntüsü Resim 5.6'daki gibidir.



Resim 5.6 Gönder düğmesine basıldıktan sonraki tarayıcı görüntüsü

İşlem sonucunu yeni bir pencerede açabilmek için target="_blank" komutu <form> etiketi içine eklenmiştir. Resim 5.4.'teki örnek kodlarda yer aldığı gibi aslında formların en fazla kullanılan iki özelliği vardır. Bunlar action ve method'dur. Bu iki özellik neredeyse bütün formlarda yer alır.

✓ Formların en fazla kullanılan iki özelliği action ve method'dur. Action form verilerinin sunucuda nereye gönderileceğini, method ise verilerin sunucuya nasıl gönderileceğini belirler.

Action özelliği formlardan alınan verinin sunucuda nerede değerlendirileceğini belirler. CGI, ASP veya PHP gibi uygulamalar form verilerini işlemeye kullanılır. Örneğin sunucuya gönderilen veriler bir veritabanına kaydedilebilir. Method özelliği kullanıcı verisinin sunucuya nasıl gönderileceğini, başka bir deyişle form verilerinin iletim biçimini belirler. Get ve post olmak üzere 2 değer vardır. Bu iki değer farkları şu şekilde özetlenebilir. Formlarda veri iletim metodu olarak get kullanıldığında form elemanının karşısında aldığı de-

ğer adres satırında gözükür. Örneğin Resim 5.4'teki örnekte isim kutusuna cem yazıp gönder düğmesine bastığımızda açılan sayfanın adres satırında “<http://www.webadres.com.tr/kayit.php?isim=cem>” şeklinde bir ifade yer alacaktır. Form içerisinde başka elemanlar yer alıyorsa, onların değeri de bu şekilde adres satırında çiftler hâlinde gözükcektir. Post metodunda ise veriler adres satırında gözükmeden arka planda sunucuya gönderilir. Get metodundaki gibi şifre türü verilerin adres satırında gözükmesi, güvenlik anlamında sorunlar yaratabilir. Ayrıca get metodunda form bilgilerini içeren adres gönderimi için belirli bir karakter sayısı sınırı vardır. Bu sınır tarayıcı ya da sunucuya göre değişebilir. Başka bir deyişle uygulamada çok yeri olmasa da get metoduyla sınırlı miktarda form verisi sunucuya gönderilebilir.

Ancak get ile veri iletiminin bir avantajı vardır. Formlarla yapılmış belirli sorgulamaların sonuçları sık kullanılanlara eklenebilir. Bu bir arama motorunda yapılan araştırma ya da bir WEB sitesinde belirli değişkenlere göre yapılan arama olabilir. Örneğin bir ikinci el araç sitesinde dizel, sedan ve mavi bir araç aradığımızda, arama sonuçlarını sık kullanılanlara ekleyerek daha sonra inceleme olanağına sahip olabiliriz.

Name özelliği formların ve form elemanlarının çoğunda yer alır. Özellikle veri taşıyan ve üzerinde sunucu ya da istemcide işlem yapılacak form elemanlarında name özelliği kullanılmalıdır. Form elemanının taşıyacağı veri için kullanılacak değişkenin adı name ile belirlenir. Örneğin `<input type="text" name="ad">` ifadesinde ad isimli bir metin kutusu oluşturulmaktadır. Kutuya girilen kullanıcı ismi Ali ise, `ad=Ali` olarak sunucuya gönderilir.



dikkat

name özelliği veri barındıran bütün form elemanlarında kullanılmalıdır. Bir form elemanındaki veriyi tutacak olan değişken adı name ile belirlenir.

Öğrenme Çıktısı



- 1 Formların kullanım amaçlarını ifade edebilme.
- 2 Genel form yapısını tanımlayabilme.
- 3 Form özelliklerinin kullanım amaçlarını ifade edebilme.

Araştır 1

Bağlantı özellikli bir metnin, yönlendirdiği sayfaya aynı zamanda bir veri teslim etmesini nasıl sağlayabileceğinizi araştırınız.

İlişkilendir

Web üzerinde karşılaştığınız bir form sayfasının HTML kodlarını görüntüleyerek, üzerindeki verileri nereye yönlendirdiğini keşfetmeye çalışınız.

Anlat/Paylaş

GET ve POST metodlarının yapısını göz önünde bulundurarak, kullanıcıların parolalarının iletimi için hangisinin daha uygun olacağını yorumlayınız.

FORM ELEMANLARI

Form elemanları genellikle `<form>...</form>` etiketleri arasında kalan alana girilen, kullanıcının veri girişi ya da seçim yaptığı birimlerdir. Örneğin bir siteye kayıt olurken mail adresimizi girdiğimiz metin kutusu veya öğrenim durumumuzu belirlediğimiz açılır listeler birer form elemanıdır. Web sayfalarında yer alabilecek önemli form elemanlarının listesi Tablo 5.2'de özetlenmiştir.

- ✓ Form elemanları, WEB sitesinde metin kutuları, açılır listeler, onay kutuları, düğmeler gibi kullanıcının veri girişi ya da seçim yaptığı alanları oluşturur.

Tablo 5.2 Form elemanları

Form Elemanı	Alt Eleman	HTML Sürüm
Input (type ile)	text (metin)	
	password (şifre)	
	tel (telefon numarası)	HTML5
	email (E-posta)	HTML5
	date (tarih)	HTML5
	month (ay)	HTML5
	week (hafta)	HTML5
	time (zaman)	HTML5
	datetime-local (yerel zaman)	HTML5
	url (WEB adresi)	HTML5
	checkbox (onay kutusu)	
	radio (radyo düğmesi)	
	submit (gönder düğmesi)	
	image (resimli düğme)	
	reset (sıfırlama düğmesi)	
	button (düğme)	
	file (dosya yükleme)	
	hidden (gizli)	
	color (renk)	HTML5
	range (aralıklı kaydırma)	HTML5
	number (numara)	HTML5
	search (arama)	HTML5
button (düğme) (type ile)	Button	
	Submit	
	Reset	
fieldset (gruplayıcı)	legend (başlık)	
select (açılır liste)	optgroup (gruplama başlığı)	
	option (seçenek)	
textarea (metin alanı)	-	
datalist (veri listesi)	-	HTML5
label (etiketler)	-	
output (sonuç çıktısı)	-	HTML5
keygen (kullanıcı kimliği doğrulama)	-	HTML5
meter (gösterge)	-	HTML5
progress (ilerleme çubuğu)	-	HTML5

Kaynak: <https://www.w3.org/wiki/HTML/Elements/form>, 2016

Görüldüğü gibi button hem input elemanın bir özelliği, hem de ayrı bir form elemanıdır. Bu durumun nedeni Button (Düğme) bölümünde anlatılmıştır. Ayrıca output, progress, meter, keygen elemanları PHP, ASP gibi WEB programlama dilleriyle birlikte kullanılabildiğinden, kitapta bu elemanlara yer verilmemiştir. Bunlardan keygen ve output elemanları, kullanıcının seçim ve giriş yapabilmesi nedeniyle progress ve meter elemanlarına göre daha fazla form özelliği taşımaktadır.



Keygen ve output elemanlarının uygulamalı örneklerini incelemek için <http://huseyinbodur.net/?p=791#more-791> adresinden faydalanabilirsiniz.

Form elemanlarından bazılarının sağ tarafında yer alan HTML5 ibaresi, bu elemanların HTML5 ile gelen yeni elemanlar olduğunu göstermektedir. Bu bölümden sonra form elemanlarının her birinin kullanım amaçlarını açıklayan, bunları örneklendiren ve kendilerine verilebilecek özellikleri gösteren bölümlere yer verilmiştir.



yaşamla ilişkilendir

Tarayıcılardaki Otomatik Form Doldurma Özelliğinde Büyük Güvenlik Açığı!

Tarayıcıların kolaylık sağlamak için sunduğu otomatik form doldurma özelliğinde kişisel verileri tehlikeye atan önemli bir güvenlik açığı tespit edildi.

Tarayıcıların çoğunda bulunan otomatik form doldurma (tamamlama) özelliğini mutlaka farketmişsinizdir. İnternet sitelerinde herhangi bir form doldururken bu özellik sayesinde formdaki boşlukları otomatik olarak tamamlayabiliyorsunuz. Ancak bu özelliği kullanırken artık iki kere düşünmeniz gerekiyor. Zira bu küçük özellik büyük bir güvenlik açığını da beraberinde getiriyor.

Anttivijlami isimli bir GitHub kullanıcısının farkettiği bu güvenlik açığı, kullanıcıların otomatik doldurma (autofill) özelliği ile kaydedilmiş

tüm bilgilerin -tek bir form alanını bu özellik ile doldursa dahi- açığa çıkmasına sebep oluyor. Kötü niyetli siber saldırı gerçekleştirmek isteyenler ise bu verilere kolayca ulaşabiliyor.

Peki ne yapmak gerekiyor?

Kişisel verilerinizin tehlikeye girmemesi için ilk yapmanız gereken -ilgili güvenlik açığı kapatılana kadar- tarayıcı ayarlarından otomatik form doldurma özelliğini kapatmak. Chrome, Firefox, Safari gibi tarayıcıların ayarlarına girerek ilgili özelliği kapatabilirsiniz. Ayrıca güvenmediğiniz internet sitelerinde form doldurmaktan da kaçınmanızı tavsiye ederiz.

Kaynak: <http://www.webmasto.com/tarayicilardaki-otomatik-form-doldurma-ozelliginde-buyuk-guvenlik-acigi>

Input (Giriş)

Input elemanı genellikle <form> etiketi içinde kullanılır ve birçok formatta veri girebilmesini sağlar. Veri girişi tipi type komutuyla belirlenir ve kapama etiketi kullanılmadan özellikler eklenir.

Input elemanlarına, kullanılan veri tipine göre çeşitli özellikler yüklenebilir. İlgili input elemanında en fazla kullanılan özelliklerin nasıl kullanılacağı ve hangi değerleri alabileceği Tablo 3'te listelenmiştir. Ayrıca Tablo 1'de yer alan autocomplete ve name özelliği de Input elemanının bir özelliği olarak kullanılabilir. Yine Tablo 1 'deki action, enctype, method, validate ve target özelliklerinin başına form eki eklenerek formaction, formenctype, formmethod, formvalidate ve formtarget şeklinde kullanılabilir. Bu durumda form elemanına, ait olduğu formun özellikleri dışında kendi özellikleri eklenebilir.



dikkat

Input elemanında kapama etiketi kullanılmaz.



Input elemanı ve bu elemanın özelliklerine ilişkin örnekleri incelemek ve uygulamalar yapmak için http://www.html5ersleri.org/index.php?getir=html5_formattr&ID=34 adresinden faydalanabilirsiniz.

Tablo 5.3 Input özellikleri

Özellik	Değer ve Kullanım	Tanım
accept	= "audio/*" ses dosyaları = "video/*" görüntü dosyaları = "image/*" resim dosyaları = ".jpg, .doc" gibi (virgülle ayrılmış)	Yüklenebilecek dosya türünü ve formatını belirler. Soldaki değerler araya virgül konarak istenilen şekilde kullanılabilir.
alt	= "Gönder" gibi alternatif değer	Resimli düğmenin resmi sayfaya yüklenemediği durumlarda metin olarak düğmenin işlevini ifade eder ve resmin yerine geçer.
checked	<i>checked</i>	Onay kutusunun varsayılan değer olarak işaretli olarak görünmesini sağlar.
height	= "20" gibi piksel cinsinden değer	Resimli düğmenin yüksekliğini piksel cinsinden belirler.
list HTML5	= "liste" gibi bir açılır liste kimliği	Önceden oluşturulmuş bir açılır listenin kimliğini kullanarak liste elemanlarına erişir.
max min	= "20" gibi sayısal değer = "1979-12-31" gibi tarih	Bir form elemanının alabileceği en yüksek veya düşük değeri belirler.
maxlength minlength	= "20" gibi sayısal değer	Bir elemanın alabileceği en yüksek değeri belirler. Pozitif tam sayı değerleri alır.
multiple HTML5	<i>multiple</i>	Kullanıcının birden fazla email girebilmesini ya da dosya yüklemelerinde birden fazla dosya seçebilmesini sağlar. E-postalar için aralarda virgül kullanılır.
pattern HTML5	= "[A-Za-z]" regexp değerler (regular expressions)	Belirli kalıplara göre veri girişi sağlar. Örneğin kullanıcı adı belirlerken büyük ve küçük harf kullanımının zorunlu tutulması. "[A-Za-z]" bu durumu sağlar.
placeholder HTML5	= "birisi@webadres.com.tr" gibi	Veri girişi öncesi ipucu sağlar. Örneğin e-posta girilecek alanda örnek bir e-posta gösterilmesi (birisi@mail.com).
readonly	<i>readonly</i>	Veri girişini kapatır. İlgili form elemanında yalnızca belirlenen değer görünür.
required HTML5	<i>required</i>	Formun doldurulması zorunlu alanlarını belirler. Bu alanlar doldurulmadan veri girişi sağlanmaz.
size	= "20" gibi sayısal değer	İlgili form elemanının kaç karakter uzunluğunda veri alabileceğini belirler.
src	= "http://ornek.com/dugme.jpg" gibi	Bir resimli düğmenin resim yolunu belirler (Resmin sunucuda bulunduğu yer).
step	= "3" = "-3" gibi sayısal değer	İlgili form elemanında yukarı ve aşağı yönlü olarak adım miktarını belirler.
width	= "20" gibi piksel cinsinden değer	Resimli düğmenin genişliğini piksel cinsinden belirler.
value	= "gonderdugmesi" gibi	Bir giriş elemanının varsayılan değerini belirler.

Kaynak: <https://www.w3.org/TR/html5/forms.html>, 2016

Pattern özelliğinde yer alan regexp (Regular Expressions, Düzenli İfadeler) terimi programlama dillerinde ve HTML'de özellikle veri girişlerinde sıklıkla kullanılmaktadır. Yandaki karekodu kullanarak, açılan sayfadan daha fazla örnek ve uygulamaya erişebilirsiniz.



Özellikler kullanıcı veri girerken ya da seçim yaparken fayda sağlamaktadır. Örneğin bir text elemanına bilgi girerken en fazla on karakter girilmesi isteniyorsa `maxlength="10"` şeklinde belirtilir. Böylelikle kullanıcı 10 karakterden fazla veri giremez. Şimdi sırasıyla bütün input elemanlarına ve özelliklerin nasıl kullanılabileceğine göz atalım.

Text (Metin Kutusu)

Tek satırlık metin girişi için kullanılır. Resim 5.7'de text için örnek kodlar yer almaktadır.

```
1 <form>
2   İsim<br>
3   <input type="text" name="ad"><br>
4   Soyisim<br>
5   <input type="text" name="soyad">
6 </form>
```

Resim 5.7 Input text örneği

İsim

Soyisim

Resim 5.8 Input Text Örneğinin Tarayıcı Görüntüsü

Resim 5.7'deki kodların sırasıyla üçüncü ve beşinci satırlarında ad ve soyad isimli birer metin kutusu oluşturulmaktadır. İsim ve Soyisim bilgileri sunucuya, ad ve soyad isimli değişkenler olarak gönderilmektedir. Text için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.8'deki gibidir.

Text elemanına autocomplete, list, maxlength, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir. Örneğin formda ad ve soyad bölümünün mutlaka doldurulması isteniyorsa `<input type="text" name="ad" required>` şeklinde required komutu eklenmelidir. Tablo 3'te görüldüğü gibi required özelliğinin bir değeri yoktur. Input etiketi içine doğrudan yazılabilir.

Password (Şifre)

Kullanıcının şifre girebilmesi için bir metin alanı oluşturur. Resim 5.9'da password için örnek kodlar yer almaktadır.

```
1 <form>
2   Şifrenizi giriniz <input type="password" name="sifre">
3 </form>
```

Resim 5.9 Input password örneği

Resim 5.9'daki kodların ikinci satırında şifre isimli bir şifre kutusu oluşturulmaktadır. Bu alana girilen karakterler güvenlik amacıyla ekranda * ya da gibi sembollerle gizlenir. Forma ait bir gönder düğmesi var ise, gönder düğmesine basıldığında şifre, şifre isimli değişken olarak sunucuya gönderilmektedir. Password için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.10'daki gibidir.

Şifrenizi giriniz

Resim 5.10 Input password örneğinin tarayıcı görüntüsü

name="ad" autocomplete="off"> şeklinde düzenlenmelidir. Tablo 3'te görüldüğü gibi autocomplete özelliğinin "on" ve "off" olmak üzere iki değeri vardır.

Password elemanına autocomplete, maxlength, name, pattern, placeholder, readonly, required, ve size özellikleri verilebilir. Örneğin önceden ziyaret edilmiş ve şifre girilmiş bir sitede, şifre alanının otomatik olarak doldurulması istenmiyorsa, kodlar `<input type="password"`

Tel (Telefon Numarası)

Kullanıcının telefon numarası girebilmesi için bir alan oluşturur. Ancak bazı tarayıcılar bu özelliği desteklemez. Resim 5.11'de tel için örnek kodlar yer almaktadır.

```

1 <form action="form.php">
2   Telefon Numaranız: <input type="tel" name="telno"><br>
3   <input type="submit">
4 </form>

```

Resim 5.11 Input tel örneği

Resim 5.11'deki kodların ikinci satırında telefon numarası girilebilecek, telno isimli bir numara alanı oluşturulmaktadır. Tel için yazılmış söz konusu kodların tarayıcıdaki görüntüsü Resim 5.12'deki gibidir. Gönder Düğmesine basıldığında telefon numarası, telno isimli değişken ile sunucuya gönderilmektedir.

Telefon Numaranız:

Gönder

Resim 5.12 Input tel örneğinin tarayıcı görüntüsü

Tel elemanına autocomplete, list, maxlength, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir. Telefon numarasının belirli bir şekilde girilmesi isteniyorsa kullanıcıya bilgi verebilmek için <input type="tel" name="telno" placeholder="0 5XX XXX XX XX"> şeklinde düzenlenebilir. Burada telefon numarasının Türkiye'deki numaralandırma biçimi gösterilmiştir. Tarayıcı görüntüsü Resim 5.13'te görülmektedir.

Telefon numaranız Gönder

Resim 5.13 Diğer bir input tel örneği

Email (E-posta)

Kullanıcının e-posta adresini girebilmesi için bir e-posta alanı oluşturur. Resim 5.14'te email için örnek kodlar yer almaktadır.

```

1 <form>
2   E-posta adresiniz: <input type="email" name="email">
3   <input type="submit">
4 </form>

```

Resim 5.14 Input email örneği

Resim 5.14'teki kodların ikinci satırında email isimli bir e-posta alanı oluşturulmaktadır. Gönder düğmesine basıldığında bu alana girilen e-posta adresi, email isimli değişken ile sunucuya gönderilmektedir. Email için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.15'teki gibidir. Email, e-postalardaki @ simgesi kontrolü yaparak verileri gönderir. Alana girilen karakterlerde @ yoksa Resim 5.15'teki tarayıcı uyarısını verir. Email elemanına autocomplete, list, maxlength, multiple, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir.

E-mail: Gönder

! Lütfen e-posta adresine bir "@" işareti ekleyin. "hcs" adresinde "@" eksik.

Resim 5.15 Input email örneğinin tarayıcı görüntüsü

Date (Tarih)

Kullanıcının tarih girebilmesi için bir alan oluşturur. Bununla birlikte Form Elemanları adlı bölümde listelenen diğer tarih ve zaman elemanları (datetime-local, month, week, time) benzer şekilde kullanıldığından kitapta bu elemanlara ayrıca yer verilmeyecektir. Resim 5.16'da date için örnek kodlar yer almaktadır.

```

1 <form action="demo.asp">
2   Doğum tarihiniz: <input type="date" name="dogumgunu">
3   <input type="submit">
4 </form>

```

Resim 5.16 Input date örneği

Doğum tarihiniz: gg . aa . yyyy

Haziran 2016

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Resim 5.17 Input date örneğinin tarayıcı görüntüsü

Resim 5.16'daki kodların ikinci satırında dogumgunu isimli bir tarih alanı oluşturulmaktadır. Date için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.17'deki gibidir. Gönder düğmesine basıldığında bu alanda seçilen tarih, dogumgunu isimli değişken ile sunucuya gönderilmektedir.

Date elemanına autocomplete, list, max, min, name, readonly, required, step ve value özellikleri verilebilir. Örneğin tarih alanına 1 Ocak 2000 tarihinden önceki tarihlerin girilmemesi isteniyorsa <input type="date" name="dogumgunu" min="2000-01-01"> şeklinde min özelliği kullanılabilir.

URL (WEB Adresi)

Kullanıcının bir WEB adresi girebilmesi için alan oluşturur. Resim 5.18'de url için örnek kodlar yer almaktadır.

```

1 <form action="sayfaniz.asp">
2   WEB adresiniz: <input type="url" name="sayfa"><br>
3   <input type="submit">
4 </form>

```

Resim 5.18 Input url örneği

Resim 5.18'deki kodların ikinci satırında sayfa isimli bir url alanı oluşturulmaktadır. URL için yazılmış bu kodların tarayıcıdaki görüntüsü Resim 5.19'daki gibidir. Gönder düğmesine basıldığında bu alana girilen WEB adresi, sayfa isimli değişken ile sunucuya gönderilmektedir. Url elemanı, girilen metinde http:// belirtecini kontrol eder. Alana girilen karakterlerde http:// belirteci yoksa Resim 5.19'daki tarayıcı uyarısını verir. Url elemanına autocomplete, list, maxlength, name, pattern, readonly, required, size, ve value özellikleri tanımlanabilir.

WEB adresiniz:

! Lütfen bir URL girin.

Resim 5.19 Input url örneğinin tarayıcı görüntüsü

Checkbox (Onay Kutusu)

Kullanıcının işaretleyebilmesi için onay kutuları oluşturur. Resim 5.20'de checkbox için örnek kodlar yer almaktadır.

```

1 <form action="demo_form.asp">
2   <input type="checkbox" name="arac1" value="bisiklet" checked> Bisiklet<br>
3   <input type="checkbox" name="arac2" value="taksi"> Taksi<br>
4   <input type="checkbox" name="arac3" value="tekne"> Tekne<br>
5   <input type="submit" value="Gönder">
6 </form>

```

Resim 5.20 Input checkbox örneği

Resim 5.20'deki kodların ikinci ve dördüncü satırları arasında arac1, arac2 ve arac3 isimli onay kutuları oluşturulmaktadır. Checkbox için yazılmış bu kodların tarayıcı görüntüsü Resim 5.21'deki gibidir.

☒ Bisiklet
☐ Taksi
☐ Tekne

Resim 5.21 Input checkbox örneğinin tarayıcı görüntüsü

Kullanıcı kendine uyan seçenek ya da seçenekleri işaretleyebilir. Gönder düğmesine basıldığında bisiklet değeri arac1, taksi değeri arac2 ve tekne değeri arac3 isimli değişkenlerle sunucuya gönderilmektedir. Checkbox elemanına checked, name, required, ve value özellikleri verilebilir. Örneğin bisiklette olduğu gibi kullanıcıya WEB sayfası açıldığında onay kutusu işaretli gösterilecekse checked özelliği eklenebilir.

Radio (Radyo Düğmesi)

Kullanıcının seçim yapabilmesi için radyo düğmeleri oluşturur. Resim 5.22'de radio için örnek kodlar yer almaktadır.

```
1 <form action="kayit.php">
2   Cinsiyetiniz<br>
3   <input type="radio" name="cinsiyet" value="bay"> Bay<br>
4   <input type="radio" name="cinsiyet" value="bayan"> Bayan<br>
5   <input type="submit" value="Gönder">
6 </form>
```

Resim 5.22 Input radio örneği

Cinsiyetiniz

☐ Bay

☐ Bayan

Resim 5.23 Input radio örneğinin tarayıcı görüntüsü

Resim 5.22'deki kodların üçüncü ve dördüncü satırlarında cinsiyet isimli radyo düğmeleri oluşturulmaktadır. Radio için yazılmış bu kodların tarayıcı görüntüsü Resim 5.23'teki gibidir.

Kullanıcı kendine uyan seçeneklerden birisini seçebilir. Gönder düğmesine basıldığında cinsiyet değişkeni bay veya bayan değerlerinden biriyle sunucuya gönderilmektedir. Radio elemanına checked, name, required, ve value özellikleri verilebilir.

Submit (Gönder Düğmesi)

Submit, bundan önceki form elemanları için verilen örneklerde görüldüğü gibi, genel olarak kullanıcının form verilerini sunucuya göndermesi için kullanılır. Resim 5.24'te submit için örnek kodlar yer almaktadır.

```
1 <form action="kayit.php">
2   Cinsiyetiniz:<br>
3   <input type="radio" name="cinsiyet" value="bay">Bay<br>
4   <input type="radio" name="cinsiyet" value="bayan">Bayan<br>
5   <input type="submit" name="gonderdugmesi" formaction="cinsiyet.php" value="Gönder">
6 </form>
```

Resim 5.24 Input submit örneği

Resim 5.24'teki kodların beşinci satırında gonderdugmesi isimli bir gönder düğmesi oluşturulmaktadır. Value değeri düğmenin üstünde beliren yazıdır. Bu örnekte Gönder olarak belirlenmiştir. Submit için yazılmış bu kodların tarayıcı görüntüsü Resim 5.25'teki gibidir. Kullanıcının cinsiyet seçimi gonderdugmesi isimli düğmeyle sunucuya gönderilmektedir. Submit elemanına formaction, formenctype, formmethod, formvalidate, formtarget, name, ve value özellikleri verilebilir.

Cinsiyetiniz

☐ Bay

☐ Bayan

Resim 5.25 Input submit örneğinin tarayıcı görüntüsü

Ancak örnekte olduğu gibi formaction özelliğiyle cinsiyet bilgisi, form etiketinde yer alan kayıt.php adresine değil, düğme içinde tanımlanan cinsiyet.php adresine gönderilmektedir. Formaction özelliği submit için, kendi veri gönderme yolunu ya da dosyasını belirlemesini sağlar.

Image (Resimli Düğme)

Image aslında bir submit elemanıdır. Genel olarak kullanıcının form verilerini sunucuya göndermesi için kullanılır. Tek farkı gönder düğmesi yerine bir resim kullanılmasıdır. Resim 26'da image için örnek kodlar yer almaktadır.

```
1 <form action="adsoyad.asp">
2   Adınız: <br>
3   <input type="text" name="isim"><br>
4   Soyadınız: <br>
5   <input type="text" name="soyisim"><br>
6   <input type="image" src="gondersimgesi.png" alt="Gönder" width="55" height="55">
7 </form>
```

Resim 5. 26 Input image örneği

Resim 5.26'daki kodların altıncı satırında image tipinde düğme oluşturulmaktadır. Type, image değişkeniyle düğme türünü (bu örnekte resimli düğme) belirlemektedir. Src resmin sunucudaki konumunu göstermektedir. Örnekte olduğu gibi resim ögesi olarak gondersimgesi.png isimli resim kullanılmıştır. Bu resmin sayfada görüntülenememesi durumunda, alt özelliğiyle sayfada tıklanabilir Gönder yazısı belirecektir. Bu yazıya basıldığında bilgi adsoyad.asp adresine gönderilecektir. Resmin piksel cinsinden eni ve boyu sırasıyla width ve height özellikleriyle belirlenmiştir.



Resim 5. 27 Input image örneğinin tarayıcı görüntüsü

Image için yazılmış bu kodların tarayıcı görüntüsü Resim 5.27'deki gibidir. Kullanıcının ad ve soyad bilgisi image elemanıya, yani mavi ok simgesiyle sunucuya gönderilmektedir. Image elemanına alt, formaction, formenctype, formmethod, formvalidate, formtarget, name, src ve value özellikleri verilebilir.

Reset (Sıfırlama Düğmesi)

Reset genel olarak çok fazla veri girilmesi gereken formlarda önceden girilmiş verilerin tarayıcı üstünde silinmesi için kullanılır. Başka bir deyişle form elemanlarına veri girilmişse, form elemanlarını sayfa yüklendiğindeki değerlerine döndürür. Ancak submit gibi düğmelerle veriler sunucuya gönderilmediği sürece, bilgiler sadece tarayıcı üstünde ilk değerlerine döner. Resim 5.28'de reset için örnek kodlar yer almaktadır.

```
1 <form action="bilgiler.php">
2   E-postanız: <input type="text" name="email"><br>
3   Adresiniz: <input type="text" name="adres"><br>
4   <input type="radio" name="ogrenimdurumu" value="ilkokul"> ilkokul<br>
5   <input type="radio" name="ogrenimdurumu" value="ortaokul" checked=""> Ortaokul<br>
6   <input type="radio" name="ogrenimdurumu" value="Diger"> Diğer<br>
7   <input type="submit" value="Gönder">
8   <input type="reset" value="Resetle">
9 </form>
```

Resim 5. 28 Input reset örneği

Resim 5.28'deki kodların sekizinci satırında type özelliğinin reset değeriyle reset tipinde düğme oluşturulmaktadır. Reset için yazılmış bu kodların tarayıcı görüntüsü Resim 5.29'deki gibidir.

E-postanız:

Adresiniz:

☐ İlkokul
☒ Ortaokul
☐ Diğer

Durum 1.
WEB sayfası açıldığında form değerleri

E-postanız:

Adresiniz:

☐ İlkokul
☐ Ortaokul
☒ Diğer

Durum 2.
Kullanıcı bilgileri girişi sonrası form değerleri

Resim 5.29 Input reset örneği tarayıcı görüntüsü

Resim 5.29 'da iki durum vardır. Kullanıcı form alanlarını doldururken yanlış veriler girmişse, resetle düğmesine basarak Durum 1'e dönebilir. Dikkat edilirse sadece e-posta ve adres bilgilerinin girildiği metin alanları değil, öğrenim durumunun belirlendiği seçim düğmeleri de ilk değerlerine dönmektedir. Reset elemanın herhangi bir özelliği yoktur.

Button (Düğme)

Button öntanımlı olmayan düğmeler oluşturur. Başka deyişle gerçekleştirecekleri eylem sayfa tasarımcısı tarafından belirlenir. Örneğin oluşturulan düğme sayfada bir javascript kodunu etkinleştirebilir. Resim 5.30'da button için örnek kodlar yer almaktadır.

```

1 <form>
2   <input type="button" value="Mesaj için tıkla" onclick="mesaj()">
3 </form>
4
5 <script>
6   function mesaj() {
7     alert("Merhaba Dünya!");
8   }
9 </script>

```

Resim 5.30 Input button örneği

Resim 5.30'daki kodların ikinci satırında type özelliğinin button değeriyle, button tipinde düğme oluşturulmaktadır. Value değeri düğme üstünde görünen ifadeyi belirlemektedir. Onclick ise altıncı ve dokuzuncu satırlar arasında yer alan mesaj isimli javascript kodunu çalıştırmaktadır. Kodların tarayıcı görüntüsü Resim 5.31'deki gibidir. Resim 5.31'de görüldüğü gibi sol tarafta sayfa içinde yer alan Mesaj için tıkla düğmesine basıldığında, tarayıcının mesaj penceresinden "Merhaba Dünya!" mesajı ekranda belirmektedir.



Resim 5.31 Input button örneğinin tarayıcı görüntüsü

Button elemanına autofocus, disabled, name ve value özellikleri verilebilir. Örneğin disabled özelliği, kullanıldığı düğme elemanını etkisizleştirerek, tıklanamaz duruma getirir.

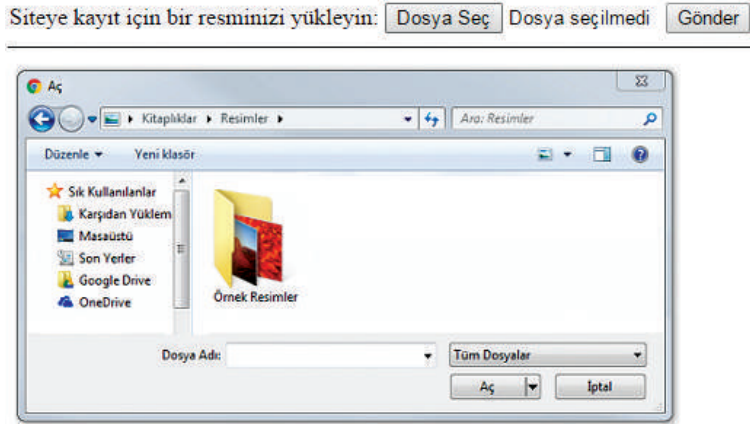
File (Dosya Yükleme)

File, kullanıcının tarayıcıda dosya/dosyalar seçebilmesi için kullanılır. Seçim yapıldıktan sonra forma ait submit düğmesiyle dosya sunucuya gönderilebilir. Resim 5.32'de file için örnek kodlar yer almaktadır.

```
1 <form action="resim.php">
2   Siteye kayıt için bir resminizi yükleyin:
3   <input type="file" name="resim">
4   <input type="submit" name="gonderdugmesi" value="Gönder">
5 </form>
```

Resim 5.32 Input file örneği

Resim 5.32'deki kodların üçüncü satırında sunucuya dosya yükleyebilmek resim isimli dosya yükleme elemanı oluşturulmaktadır. Resim 5.33'te kullanıcının resim yükleyebilmesi için oluşturulan formun tarayıcı görüntüsü yer almaktadır.



Resim 5.33 Input file örneğinin tarayıcı görüntüsü

File tarayıcıya göre farklı isimlerde dosya seçim düğmesi oluşturur. Resim 5.33'te Google Chrome'da Dosya Seç düğmesinin kullanıldığı görülmektedir. İnternet Explorer'da ise bu düğmenin adı Gözet'dir. Kullanıcı dosya seç düğmesine bastığında dosya seçim penceresi belirir. Resim seçildikten sonra Gönder düğmesiyle resmi sunucuya göndermektedir.

✓ Input file ile yaratılan ve dosya seçme penceresini açmak için kullanılan düğme, tarayıcılarda farklı isimlerde gösterilir. Örneğin Google Chrome'da Dosya Seç, İnternet Explorer'da ise Gözet olarak sayfada belirir.

File elemanına accept, multiple, name, required ve value özellikleri verilebilir. Örneğin birden fazla dosyanın seçilebilmesi isteniyorsa <input type="file" name="resim" multiple> şeklinde multiple özelliği eklenebilir.

Hidden (Gizli)

Hidden, WEB sayfasındaki formlarda gerekli ancak kullanıcının görmemesi gereken ya da görmesinin bir anlam taşımadığı verileri gizlemek için kullanılır. Formda yer alan gizli veriler de sunucuya gönderilir. Bu elemanı kullanabilmek için javascript ve WEB programlamasındaki dillerden yararlanılmaktadır. Resim 5.34'te hidden için örnek kodlar yer almaktadır.

```

1 <form action="takim.asp">
2   Tuttuğunuz takım: <input type="text" name="takim"><br>
3   <input type="hidden" name="ulke" value="Türkiye">
4   <input type="submit" value="Gönder">
5 </form>

```

Resim 5.34 Input hidden örneği

Resim 5.34'teki kodların üçüncü satırında ulke isminde bir gizli eleman oluşturulmaktadır. Hidden için yazılmış örnek kodların tarayıcı görüntüsü Resim 5.35'teki gibidir. Bu örnekte kullanıcının tuttuğu takımı girebilmesi için bir form hazırlanmıştır. Ancak formda, kullanıcının ülkesini kendisinin girebileceği bir alan yoktur. Bunun yerine ülke bilgisi, IP ya da tarayıcı bilgileri kullanılarak otomatik olarak form verisine eklenebilir. Kullanıcı gönder düğmesine bastığında hem tuttuğu takım hem de yaşadığı ülke sunucuya gönderilebilir. Hidden elemanına name ve value özellikleri verilebilir.

Tuttuğunuz takım:

Resim 5.35 Input hidden örneğinin tarayıcı görüntüsü

Color (Renk)

Color kullanıcının bir renk seçebilmesi için kullanılır. Seçilen renk, rengin sayısal ifadesi olarak sunucuya gönderilir. Resim 5.36'da color için örnek kodlar yer almaktadır.

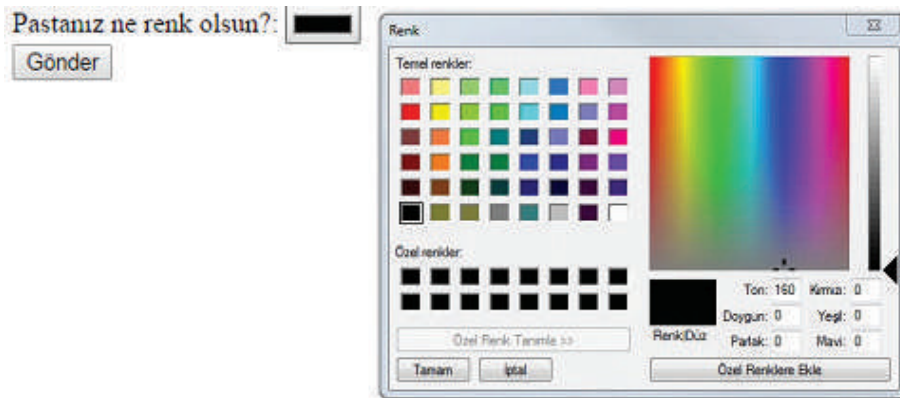
```

1 <form action="siparis.asp">
2   Pastanız ne renk olsun?: <input type="color" name="pastarengi"><br>
3   <input type="submit">
4 </form>

```

Resim 5.36 Input color örneği

Resim 5.36'daki kod örneğinin ikinci satırında pastarengi isminde bir renk seçim düğmesi oluşturulmaktadır. Kullanıcının pasta siparişi için renk belirleyebilmesini sağlayan kodların tarayıcı görüntüsü Resim 5.37'deki gibidir. Kullanıcı Pastanız ne renk olsun? yazısının yanındaki renk seçim düğmesine tıkladığında resimde görülen bir renk paleti belirlemektedir. Kullanıcı istediği rengi bu paletten seçebilmektedir.



Resim 5.37 Input color örneğinin tarayıcı görüntüsü

Color elemanına autocomplete, list, name ve value özellikleri verilebilir. Örneğin resimde gönderilen renk seçim düğmesinin renginin kırmızı olması isteniyorsa `<input type="color" name="pastarengi" value="#ff0000">` şeklinde value özelliği kullanılabilir.



WEB tasarımında ve programlamada renkler sayısal olarak ifade edilirler. Renklerin sayısal değerleri genellikle onaltılık (Hexadecimal) sistemle gösterilir. Konu ile ilgili daha fazla bilgi ve örneği içeren WEB sayfasına erişim için yandaki karekodu kullanabilirsiniz.

Range (Aralıklı Kaydırma)

Kullanıcının sayı değeri belirleyebilmesi için bir kaydırma çubuğu oluşturur. Belirlenen değer bir form verisi olarak sunucuya gönderilir. Resim 5.38'de range için örnek kodlar yer almaktadır.

```
1 <form action="siparis.asp">
2   Ürün sayısı? 1<input type="range" name="adet" value="1" min="1" max="5">5
3   <input type="submit" value="Ürünü onayla">
4 </form>
```

Resim 5.38 Input range örneği

Resim 5.38'deki kodların ikinci satırında adet isminde bir aralıklı kaydırma elemanı oluşturulmaktadır. Kullanıcının aynı üründen kaç adet sipariş vereceğini belirleyen formda, değer adet değişkeniyle sunucuya gönderilmektedir. Ürün sayısı aralıklı kaydırma elemanı ile belirlendikten sonra, ürünü onayla isimli submit düğmesiyle ürün sayısı sunucuya iletilmektedir. Min kaydırma çubuğundaki en küçük değeri, max en büyük değeri belirler. Value değeri sayfa açıldığında kaydırma çubuğu işaretçisinin nerede (Hangi değerde)



Resim 5.39 Input range örneğinin tarayıcı görüntüsü

olacağını belirler. Bu örnekte en küçük değer olan 1 değeri verilmiştir. Kodların tarayıcı görüntüsü Resim 5.39'daki gibidir. Range elemanına autocomplete, disabled, form, list, max, min, name, step, time ve value özellikleri verilebilir.

Number (Sayı)

Kullanıcının değer belirleyebilmesi ya da girebilmesi için bir sayı alanı oluşturur. Bu alan içinde yukarı ve aşağı oklar bulunur. Böylelikle kullanıcı girdiği ya da sayfada beliren değeri yukarı veya aşağı değiştirebilir. Belirlenen değer bir form verisi olarak sunucuya gönderilir. Resim 5.40'ta number için örnek kodlar yer almaktadır.

```
1 <form action="adet.asp">
2   Bir sayı girin:<br>
3   <input type="number" value="4" name="sayi" min="1" max="10" step="2">
4   <input type="submit">
5 </form>
```

Resim 5.40 Input number örneği

Resim 5.40'taki kodların üçüncü satırında sayi isimli bir sayı alanı oluşturulmaktadır. Belirlenen ya da kutuya girilen değer sayi değişkeniyle sunucuya gönderilmektedir. Min sayı alanındaki en küçük değeri, max en büyük değeri belirler. Value değeri sayfa açıldığında sayı alanında hangi sayının olacağını belirler. Bu örnekte 4 değeri verilmiştir. Step ise değerlerin kaçar kaçar artacağını ya da azalacağını belirler. Resim 5.41'de kodların tarayıcı görüntüsü yer almaktadır. Number elemanına autocomplete, disabled, form, list, max, min, name, placeholder, readonly, required, step, ve value özellikleri verilebilir.



Resim 5.41 Input number örneğinin tarayıcı görüntüsü

Search (Arama)

Kullanıcının arama yapabileceği bir alan oluşturur. Aslında bu eleman İnternet üzerinden ya da diğer aramalar için gerekli özelliklere sahip değildir. Mevcut hâliyle yapabildiği tek şey, alana girilen değeri bir arama değeri olarak sunucuya göndermek, kullanıcı bilgisayarında gerekli WEB programlama dilleriyle çeşitli aramalar yaptırmaktır. Ayrıca x simgesiyle arama satırındaki mevcut değeri siler. Resim 5.42'de search için örnek kodlar yer almaktadır.



dikkat

tel (Telefon Numarası) elemanında olduğu gibi search de tarayıcılar tarafından desteklenmez.

```
1 <form action="arama.asp">
2   Ara: <br>
3   <input type="search" name="aramayap">
4   <input type="submit">
5 </form>
```

Resim 5.42 Input search örneği

Ara:

Resim 5.43 Input search örneğinin tarayıcı görüntüsü

Resim 5.42'deki kodların üçüncü satırında aramayap isimli bir arama elemanı oluşturulmaktadır. Alana girilen değer aramayap değişkeniyle sunucuya gönderilmektedir. Kodların tarayıcıdaki görüntüsü Resim 5.43'teki gibidir. Search elemanına autocomplete, list, maxlength, name, pattern, placeholder, readonly, required, size ve value özellikleri verilebilir.

Button (Düğme)

Button daha önceki input bölümünde bahsedilen, <form> etiketi dışında da kullanılabilen genel amaçlı düğme türleri yaratır. Button için kullanılabilecek özellikler Tablo 4'te özetlenmiştir.

Tablo 5.4 Button özellikleri

Özellik	Değer ve Kullanım	Tanım
type	= "button" = "submit" = "reset"	Ne tür düğme yaratılacağını belirler.
autofocus HTML5	autofocus	Sayfa açıldığında kullanıcıyı ilgili düğmeye yönlendirir.
disabled	disabled	Düğmeyi tıklanamaz duruma getirir. Düğmenin etkisiz olmasını sağlar.
form HTML5	= "formkimligi" gibi	İlgili formun kimliğini kullanarak, o form için işlem yapılacağını belirler.
formaction Submit için HTML5	= "http://ornek.com/kayit.php" gibi	Form verilerinin formun kendi yolu dışında başka bir yola gönderilmesini sağlar.
formenctype Submit için HTML5	= "application/x-www-form-urlencoded" = "multipart/form-data" = "text/plain"	Form verileri sunucuya gönderilirken nasıl kodlanacağını belirtir.
formmethod Submit için HTML5	= get = post	Form verilerinin formun kendi iletim metodu dışında bir yöntemle gönderilmesini sağlar.

formvalidate Submit için HTML5	<i>formvalidate</i>	Formun kendi veri doğrulama özelliğini devre dışı bırakarak kendi değerine göre doğrulama işlemi yapar.
formtarget Submit için HTML5	=_blank (Yeni boş sayfa) =_self (Formun açıldığı sayfa) =_parent (İlk açılmış sayfa) =_top (En üstteki sayfa)	Form kendi gösterim özelliğini devre dışı bırakarak cevabın hangi sayfada gösterileceğini belirtir. Sayfada (Boş, açık olan, ana sayfa ya da en üstteki WEB sayfası)
value	= <i>"gonderdugmesi"</i> gibi	Bir giriş elemanının varsayılan değerini belirler.
name	= <i>"resetle"</i> gibi	Düğmelere isim vermek için kullanılır.

Kaynak: <https://www.w3.org/TR/html-markup/button.html>, 2016)

Button elemanının type özelliği, oluşturulacak olan düğmenin türünü belirlemektedir. Submit, reset ve button olmak üzere üç çeşit button vardır. Tablo 4'te görüldüğü gibi formaction, formenctype, formmethod, formvalidate ve formtarget özellikleri sadece submit tipi için kullanılabilir. Button elemanının input ile yaratılan button ile işlevleri aynıdır. Ancak input buttondan farklı olarak *<button özellikler>* şeklinde kullanılır ve HTML kapama etiketi *</>* ile sonlandırılmaları gerekir.

Bununla birlikte button elemanı ile yaratılan düğmelerin en büyük özelliği resim ve metinlerle şekillendirilebilmeleridir. Özellikle CSS (Cascading Style Sheet) ile biçimlendirilme konusunda oldukça kullanışlıdır. Resim 5.44'te submit, reset ve button türünde hazırlanmış button elemanlarına ait örnek yer almaktadır.



dikkat

Button (düğme), input ile oluşturulan düğmeden farklı olarak *<button>* etiketiyle başlatılıp, klasik HTML kodlarında olduğu gibi bitirme etiketi olan *</button>* komutuyla sonlandırılır.



CSS ile button şekillendirme örneklerini incelemek için <http://fatihhayrioglu.com/css-ile-buton-yapmak/> adresinden faydalanabilirsiniz.

```

1 <form action="hesap.php" method="post">
2   <p>
3     Adınız: <br><input type="text" name="firstname"><br>
4     Soyadınız: <br><input type="text" name="lastname"><br>
5     Eposta adresiniz: <br><input type="text" name="email"><br>
6   <br>
7   <button name="submit" type="submit" value="gonder">
8     Gönder
9   </button>
10  <button name="reset" type="reset" value="resetle">
11    Sıfırla
12  </button>
13  <button name="merhaba" type="button" value="merhaba" onclick="mesaj()">
14    Merhaba de
15  </button>
16  </p>
17 </form>
18
19
20 <script>
21 function mesaj () {
22   alert("Merhaba");
23 }
24 </script>

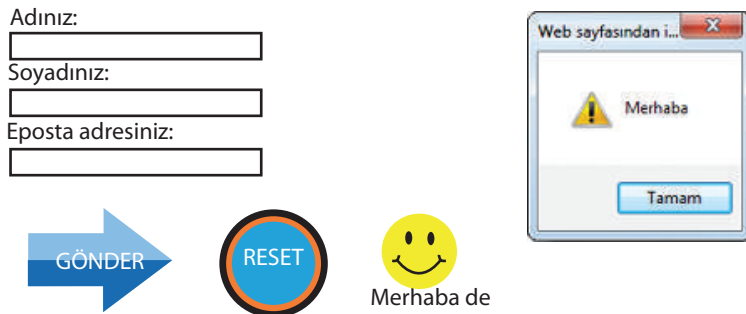
```

Resim 5.44 Submit reset button örneği

Resim 5.44'te yedinci ve dokuzuncu satırlar arasında submit isimli düğme tanımlanmıştır. Sekizinci satırda HTML'nin resim ekleme etiketi olan kullanılarak bu düğmeye jpg uzantılı gonder isimindeki resim eklenmiştir. Böylelikle tarayıcıda bir düğme yerine bu resim gözükmektedir. Resmin boyutuna bağlı olarak düğmenin genişliği width="70", yüksekliği ise height="55" kodlarıyla 70 ve 55 piksel olarak ayarlanmıştır.

Benzer şekilde onuncu ve on ikinci satırlar arasında tanımlı reset isimli düğmeye, gif uzantılı reset isimindeki resim eklenmiştir. Bu düğmenin ise yüksekliği ve genişliği reset resminin şekline göre 55 piksel olarak belirlenmiştir. Düğmeye gif uzantılı reset isimindeki resim eklenmiştir.

Merhaba de düğmesi ise, onüç ve onbeşinci satırlar arasında, merhaba.gif isimli resim dosyasından görüntüsünü almaktadır. 55 piksel yüksekliğinde ve genişliğindedir. Kodların tarayıcı görüntüsü Resim 5.45'teki gibidir.



Resim 5.45 Submit reset button örneğinin tarayıcı görüntüsü

Düğmelerin işlevleri input button ile yaratılan düğmelerle aynıdır. Gönder düğmesi kullanıcı bilgilerini sunucuya göndermektedir. Reset, Adınız, Soyadınız ve E-posta adresiniz isimli metin kutularındaki bilgileri silmekte ya da bir başlangıç değeri varsa o değere döndürmektedir. Merhaba de düğmesi ise, Resim 5.45'te görüldüğü gibi ekrana Merhaba yazan pencereyi açan (20. satırdan itibaren) javascript kodunu çalıştırmaktadır.

Fieldset ve Legend (Gruplayıcı ve Başlık)

Fieldset form elemanlarını görsel açıdan gruplamak için kullanılır. Fieldset içindeki elemanlar bir kutu içinde gözükmürler. Legend ise gruplara başlık vermek için kullanılır. Resim 5.46'da fieldset ve legend için örnek kodlar yer almaktadır.

```

1 <fieldset>
2   <legend>Kişisel Bilgiler</legend><br/>
3   Adı: <input name="ad" type="text">
4   Soyadı: <input name="soyad" type="text"><br/><br/>
5 </fieldset>
6 <fieldset>
7   <legend>İletişim bilgileri</legend><br/>
8   Adresi: <input name="personal_address" type="text" size="55" tabindex="3"/><br/>
9   Telefon No: <input type="tel" name="telno">
10 </fieldset>

```

Resim 5.46 Fieldset legend örneği

Resim 5.46'daki kodların birinci ve beşinci satırlarında ad ve soyad elemanlarından bir grup oluşturmak için fieldset kullanılmıştır. Bu gruba ikinci satırdaki legend elemanı ile kişisel bilgiler başlığı verilmiştir. Aynı şekilde altı ve onuncu satırlardaki fieldset adres ve telefon elemanlarından bir grup oluşturmak için kullanılmıştır ve yedinci satırdaki legend ile bu gruba İletişim bilgileri başlığı verilmiştir. Kodların tarayıcı görüntüsü Resim 47'deki gibidir. Fieldset elemanına disabled, form, name özellikleri verilebilir.

Resim 5.47 Fieldset legend örneğinin tarayıcı görüntüsü

Select, Optgroup, Option (Açılır Liste, Gruplama Başlığı, Seçenek)

Select açılır liste oluşturmak, option açılır liste elemanlarını belirlemek, optgroup ise gerekli durumlarda bu elemanları gruplandırmak için kullanılır. Resim 5.48'de select, optgroup ve option için örnek kodlar yer almaktadır.

```

1  Gitmek için bir ülke seçin...
2  <select name="ulkelistesi" form="ulkeler">
3    <optgroup label="Avrupa Ülkeleri">
4      <option value="1">Almanya</option>
5      <option value="2">Fransa</option>
6      <option value="3">İtalya</option>
7    </optgroup>
8    <optgroup label="Amerika Ülkeleri">
9      <option value="4">ABD</option>
10     <option value="5">Meksika</option>
11     <option value="6">Brezilya</option>
12   </optgroup>
13   <optgroup label="Asya ülkeleri">
14     <option value="7">Hindistan</option>
15     <option value="8">Çin</option>
16     <option value="9">Afganistan</option>
17   </optgroup>
18 </select>
19
20 <form action="ulke.php" id="ulkeler">
21   isim:<input type="text" name="ad">
22   <input type="submit">
23 </form>

```

Resim 5.48 Select optgroup option örneği

Resim 5.48'deki kodların ikinci ve on sekizinci satırlarında ülkeler ve bu ülkelerin yer aldığı kıtaya göre gruplandırıldığı ulkelistesi adında bir açılır liste oluşturulmaktadır. Açılır listede Avrupa, Amerika ve Asya olmak üzere ülkeler üç grupta listelenmiştir. Gruplama işlemi <optgroup> etiketiyle yapılmış ve her gruptan önce kullanılmıştır. Ayrıca bu grupların başlıklarını oluşturmak için <optgroup> etiketinin label özelliği kullanılmıştır.

Yirminci satırdan itibaren kimliği (id'si) ülkeler olan form bulunmaktadır. Görüldüğü gibi açılır liste bu formun dışındadır. Liste elemanlarını forma dâhil edebilmek için listenin form özelliğine, formun kimliği verilmiştir. Kullanıcı açılır listeden bir eleman seçip gönder düğmesine bastığında, seçilen değer sunucuya bu listeden form aracılığıyla gönderilir. Select elemanı doğrudan bir form içinde de yaratılabilir. Böylelikle select o formun elemanı olur. Kodların tarayıcı görüntüsü Resim 5.49'daki gibidir. Kullanıcı açılır listeden bir ülke seçip gönder düğmesine bastığında seçilen ülke sunucuya gönderilir. Select elemanına autofocus, disabled, form, name, required, size özellikleri verilebilir.

Resim 5.49 Select optgroup option örneğinin tarayıcı görüntüsü

Textarea (Metin Alanı)

Textarea kullanıcının yazı yazabileceği çok satırlı metin kutuları oluşturur. Satır sayısı rows değişkeniyle belirlenir. Resim 5.50'de textarea için örnek kodlar yer almaktadır.

```

1 Gezi izlenimlerinizi burada bizimle paylaşın
2 <textarea rows="5" cols="50" form="gezi">
3 Buraya düşüncelerinizi yazınız
4 </textarea>
5
6 <form action="ulke.php" id="gezi">
7   <input type="submit">
8 </form>

```

Resim 5.50 Textarea örneği

Resim 5.50'deki kodların ikinci ve dördüncü satırları arasında textarea ile bir metin alanı oluşturulmuştur. Metin alanına rows="5" kodu ile beş satır yükseklik, "cols="50" komutu ile elli karakter alabilecek genişlik verilmiştir. Altıncı ve sekizinci satırlar arasındaki kodlar ise metin alanına girilen bilgileri sunucuya göndermek için kullanılmıştır. Bu örnekte de textarea, kendi form özelliği olarak, formun kimliği (id'sini) yani "gezi" değerini kullanmaktadır. Böylelikle bu metin alanı, gezi kimlikli formun elemanı olmuştur. Kullanıcı metin alanına düşüncelerini girip Gönder düğmesine bastığında, form bilgileri sunucuya gönderilmektedir. Textarea, bu örnekten farklı olarak, doğrudan bir formun elemanı olarak bir form içinde yaratılabilir. Böylelikle o formun elemanı olur. Örnekteki kodların tarayıcı görüntüsü Resim 5.51'deki gibidir. Textarea elemanına autofocus cols, disabled, form, name, placeholder, readonly, required, rows özellikleri verilebilir.

Gezi izlenimlerinizi burada bizimle paylaşın

Buraya düşüncelerinizi yazınız

Gönder

Resim 5.51 Textarea örneğinin tarayıcı görüntüsü

Datalist (Veri Listesi)

Datalist diğer elemanlar için listelenmiş veri sağlar. Bu yönüyle select elemanına benzer. Ancak kullanıcı liste elemanlarına doğrudan göremez. Input elemanı olan list gibi elemanlarla bu elemanlara ulaşılabilir ve kullanıcılara liste elemanları gösterilebilir. Resim 5.52'de datalist için örnek kodlar yer almaktadır.

```

1 Gittiğiniz şehir:<br>
2 <input type="text" name="sehir" list="sehirismi" form="gezi">
3
4 <datalist id="sehirismi">
5   <option value="Boston">
6   <option value="New York">
7 </datalist>
8
9 <form action="sehir.asp" id="gezi">
10   <input type="submit">
11 </form>

```

Resim 5.52 Datalist örneği

Resim 5.52'de dördüncü ve yedinci satırlarda datalist kullanılarak sehirismi kimliğini (id) taşıyan bir veri listesi oluşturulmuştur. Bu veri listesinde Boston ve New York olmak üzere iki değer yer almaktadır. İkinci satırda bu veri listesindeki şehir isimlerini görüntüleyen sehir isimli bir metin alanı oluşturulmuştur. Metin kutusu içinde veri listesindeki şehirleri görüntüleyebilmek için, input text elemanının list özelliğine, datalist elemanının kimliği, yani sehirismi değeri verilmiştir. Normal görünümde sayfa yüklendiğinde aç-

lır liste elemanları kullanıcıya görünmemektedir. Liste elemanları, fare imleci metin kutusunun üzerine götürüldüğünde ortaya çıkar. Datalist aynı zamanda bir form elemanı olarak list elemanı ile birlikte formun içinde yaratılabilir. Bununla birlikte metin kutusuna tıklanıldığında açılır liste elemanlarından farklı değerler de bu metin kutusu içine girilebilir. Resim 5.53'te kodların tarayıcı görüntüsü yer almaktadır.

✓ Datalist hem kullanıcıların seçim yapabilmeleri için bir açılır liste hem de seçim listesi elemanlarından farklı değerler girebilmeleri için bir metin kutusu olarak iş görür.

Resim 5.53 Datalist örneğinin tarayıcı görüntüsü

Örnekte veri listesi ve metin kutusu yanında dokuzuncu ve on birinci satırlarda kimliği (id) gezi olan bir form tanımlanmıştır. Input text elemanının form özelliğinde bu kimliğin kullanılması, onun bu forma ait olduğunu gösterir. Kullanıcı şehir seçip veya başka bir şehir girdiğinde, formdaki gönder düğmesi girilen ya da seçilen şehir ismini sunucuya göndermektedir. Datalist elemanına verilebilecek bir özellik yoktur.

Label Elemanı

Label elemanı diğer elemanları etiketlemek için kullanılır. Sadece ilişkilendirildiği form elemanı ile bir bütün olarak, o form elemanının özelliğini kazanır. Resim 5.54'deki label elemanı için hazırlanmış örnek kod ve tarayıcıdaki görüntüsü, bu durumu daha iyi anlayabilmek için yardımcı olacaktır.

```

1 <form action="vergi.asp">
2   <label for="binek">taksi</label>
3   <input type="radio" name="arac" id="binek" value="taksi"><br>
4
5   ticari<input type="radio" name="arac" id="ticari" value="otobüs"><br>
6   <input type="submit" value="Gönder">
7 </form>

```

Resim 5.54 Label örneği

Resim 5.54'te üçüncü satırda arac isimli radyo düğmelerinden ilki için, kimliği (id'si) binek olan seçenek oluşturulmuştur. İkinci satırda ise bu radyo düğmesi için taksi etiketi (label) oluşturulmuştur. Dikkat edilirse kimliği binek olan radyo seçeneğinin id değeri ile label etiketinin for değeri aynıdır. Bu durum binek kimliği taşıyan radyo düğmesinin başına, taksi yazısının (etiketinin) gelmesini sağlamıştır. Böylelikle yazı ve düğme bir bütün olmuştur. Ancak beşinci satırda yer alan arac isimli radyo düğmelerinden ikincisi için oluşturulan ticari düğmesi için aynı durum söz konusu değildir. Ticari düğmesinin başında yer alan ticari ibaresi sadece bir metindir ve bir etiket değildir. Farklı özellikler taşıyan bu radyo düğmeleri için kullanıcı açısından uygulamada fazla fark yoktur. Tek fark taksi yazısına tıklanıldığında radyo düğmesinin işaretlenebilmesidir. Ancak ticari yazısına tıklanıldığında, yazı ile düğme ilişkilendirilmediği için düğme işaretlenemez. Label elemanına verilebilecek bir özellik yoktur.

Resim 5.55 Label örneğinin tarayıcı görüntüsü

Öğrenme Çıktısı



- 4 Form elemanlarını ve HTML5 ile gelen form elemanlarını listeleyebilme.
- 5 HTML5 kavramını açıklayabilme.
- 6 Birden fazla form elemanı ile WEB formu tasarlayabilme.
- 7 Form elemanlarının özelliklerini kullanabilme.

Araştır 2

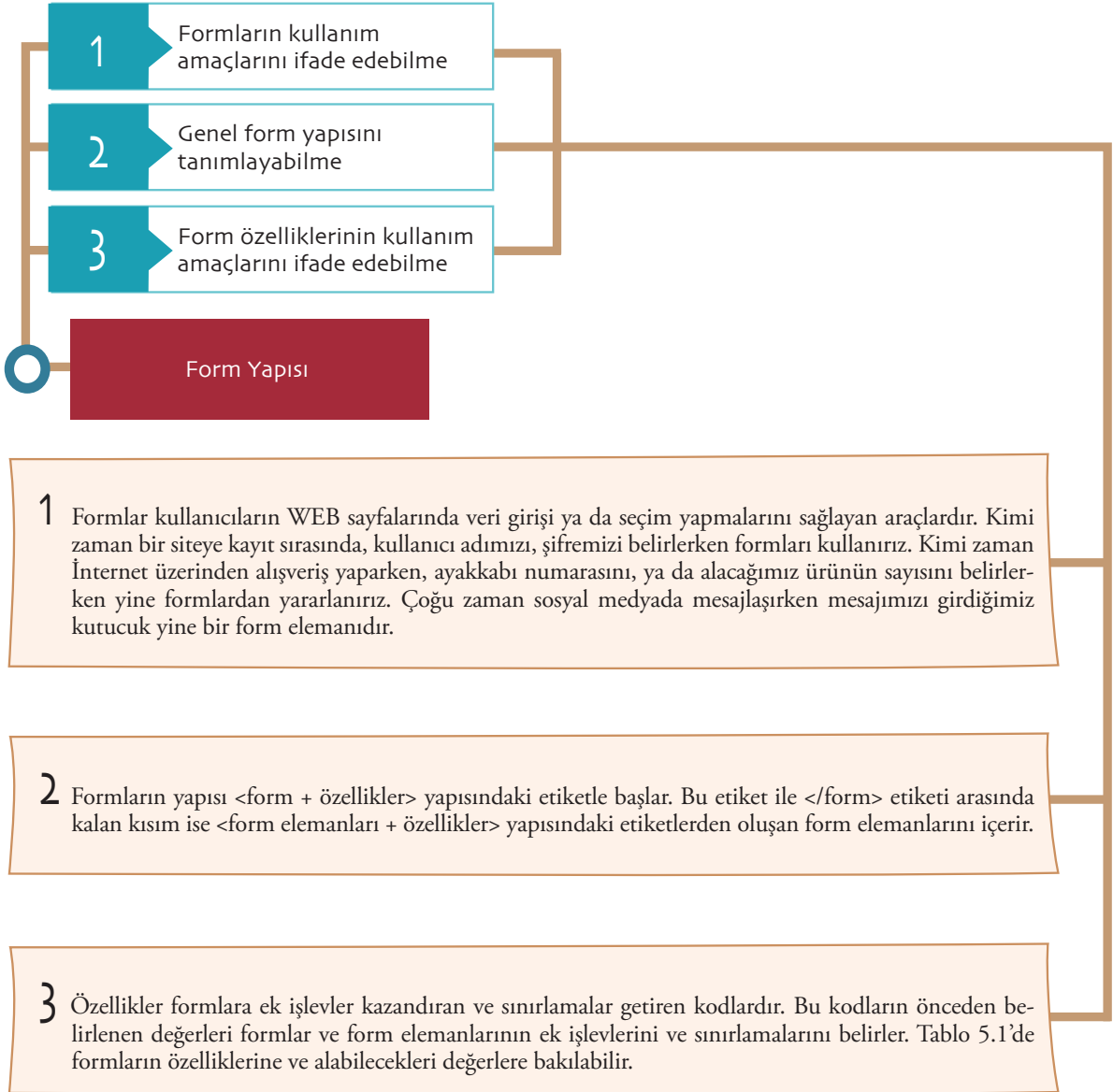
Bir e-posta alanına birden fazla e-posta adresi girilebilmesi için Resim 5.14.'de yer alan email elemanına hangi özelliğin eklenmesi gerektiğini bulunuz.

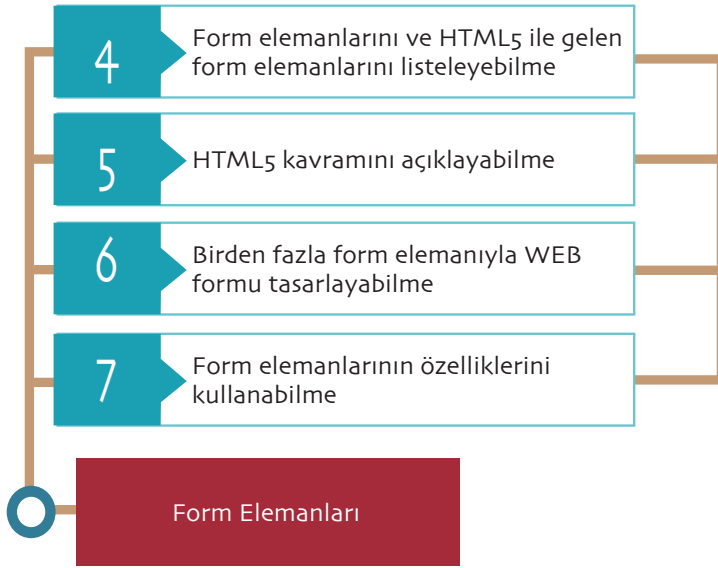
İlişkilendir

Şimdiye kadar ziyaret ettiğiniz web sayfalarında ne tür form elemanlarıyla karşılaşmış olduğunuzu hatırlamaya çalışınız.

Anlat/Paylaş

“Button” tipli form elemanı ile “Submit” tipli form elemanının farklarını tartışınız.





4 Başlıca form elemanları input (giriş), button (düğme), fieldset (gruplayıcı), select (Açılır liste), textarea (Çok satırlı metin alanı), datalist (Veri listesi-HTML5), label (Etiket), output (Sonuç çıktısı-HTML5), progress (İlerleme çubuğu-HTML5), meter (gösterge- HTML5) ve keygen (kullanıcı kimliği doğrulama- HTML5) olarak listelenebilir.

Listedeki input (giriş) elemanı type özelliği ile diğer form elemanlarından farklı olarak birçok veri giriş ve seçim alanı oluşturabilir. Bunlar text (metin), password (şifre), tel (telefon numarası-HTML5), email (E-posta-HTML5), date (tarih-HTML5), month (ay-HTML5), week (hafta-HTML5), time (zaman-HTML5), datetime-local (yerel zaman-HTML5), url (WEB adresi-HTML5), checkbox (onay kutusu), radio (radyo düğmesi), submit (gönder düğmesi), image (resimli düğme), reset (sıfırlama düğmesi), button (düğme), file (dosya yükleme), hidden (gizli), color (renk-HTML5), range (aralıklı kaydırma-HTML5), number (numara-HTML5) ve search (arama- HTML5) şeklindedir.

5 Görüldüğü gibi bazı form elemanlarının yanında HTML5 ifadesi kullanılmaktadır. Bunun sebebi HTML'nin bir WEB sayfası oluşturma dili olarak zaman içinde geliştirilmesidir. Böylelikle HTML hem kullanıcıların hem de WEB sayfası tasarımcılarının işlerini kolaylaştıracak ek işlevler kazanmaktadır. Bu kapsamda son olarak HTML5 olarak adlandırılan yükseltme sürümüne ait kodlar, etiketler ve özellikler kullanılmaktadır.

6 Bir Web sitesinde kullanıcının veri girmesi, seçimler yapması ve bunları sunucuya göndermesi için birden fazla form elemanı birlikte kullanılabilir. Tasarımcının toplamak istediği bilgiler için kullanacağı form elemanlarının nasıl kullanılacağı ilgili form elemanı bölümünden bakılabilir. Örneğin bir alışveriş sitesinde müşteri bilgilerinin toplanacağı bir form tasarlanabilir. Form elemanları olarak ad, soyad bilgileri için text, telefon bilgileri için tel, adres bilgileri için textarea gibi elemanlar kullanılabilir. Müşterinin alışveriş sırasında seçeceği ürünün sayısı number elemanı ile belirlenebilir. Renk seçiminde color elemanı kullanılabilir. Müşteri tarafından girilen bütün bu bilgiler gönder düğmesiyle sunucuya gönderilir.

7 Form elemanlarının özellikleri, elemanlara ek işlevler kazandırmak ve sınırlamalar getirmek için kullanılır. Hangi elemana hangi özelliğin verilebileceği ilgili form elemanı bölümünde yer almaktadır. Örneğin multiple özelliği file ve email elemanlarında kullanılır. File elemanında kullanıldığında kullanıcı bir siteye resim yüklerken birden fazla resmi aynı anda seçebilir. Böylelikle büyük bir kolaylık elde etmiş olur. Email elemanında ise birden fazla kişiye tek seferde mail adresi girilerek mail gönderilebilir.