

HTML5



Özet

HTML 5, HTML standardının en son sürümüdür ve eski sürümlere göre birçok yeni özellik ve iyileştirme sunar. Daha esnek ve uyarlabilir olacak ve geliştiricilerin daha karmaşık ve etkileşimli web uygulamaları oluşturmalarını sağlayacak şekilde tasarlanmıştır. Web sayfaları veya uygulamaları oluşturmak veya sürdürmek istiyorsanız, bu avantajlardan yararlanmak için HTML 5 kullanmayı düşünmelisiniz.

İçindekiler

1. HTML5 Nedir.....	3
2.HTML5'in Yapısı.....	6
3.Class ve Id Kullanımı.....	19
3.Semantik HTML.....	21
4.HTML5 ile Değişenler.....	23
5.Yeni Api'ler.....	39

HTML5 Nedir ?

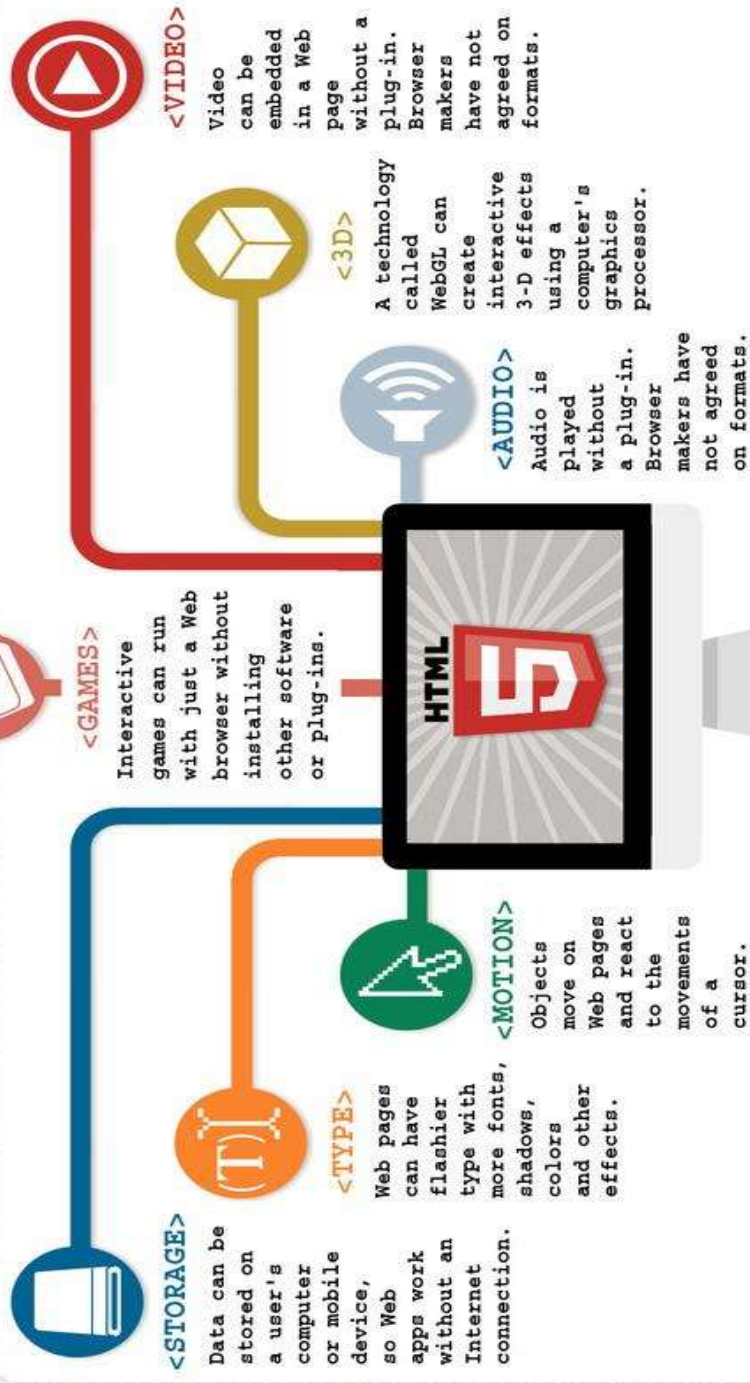
HTML5, İnternet'te içerik oluşturma ve sunma için kullanılan popüler bir işaretleme dili olan HTML'in en son sürümüdür. Dünya İnternet Konsorsiyumu (W3C) tarafından geliştirilmiş ve 2014 yılında yayınlanmıştır.

HTML5, kullanıcılar için farklı cihazlarda ve platformlarda daha standardize ve tutarlı bir deneyim sağlamak amacıyla geliştirilmiştir. HTML5, yeni elementler, öznitelikler (attributes), ses ve video oynatma, 2D ve 3D grafikler ve çevrimdışı destek gibi yeni API'ler sunmaktadır.

HTML5, ayrıca, web sayfalarının anlamını daha iyi anlamaya yardımcı olacak daha semantik bir yaklaşım benimser. Bu dil, web geliştiricileri için önemli bir adımdır ve herhangi bir web geliştirme projesi için değerlendirilmesi gereken bir seçenektir.

What Does HTML5 Do?

Key features of the next Web programming standard.



Faydaları nelerdir?

- **Daha standartlaştırılmış ve tutarlı bir kullanıcı deneyimi:** Farklı cihazlar ve platformlar arasında daha tutarlı bir deneyim sağlar.
- **Daha anlamlı ve açık bir işaretleme:** Daha anlamlı ve açık bir işaretleme kullanarak web sayfalarının elementlerin anlamını daha iyi anlamaya yardımcı olur.
- **Gelişmiş multimedya desteği:** Audio ve video içeriğine daha iyi destek sağlar. Yeni "audio" ve "video" öğeleri, web sayfalarına multimedya içeriği eklemek için kolaylaştırır.
- **Çevrimdışı web uygulamalarına destek:** Web uygulamalarının kullanıcının cihazına yerel olarak veri saklayarak, internet bağlantısı olmadan bile erişilebilir hale gelmesini sağlar.
- **Daha iyi arama sonuçları:** Semantik HTML5 kullanımı, web sayfalarının arama motorları tarafından daha kolay anlaşılmasını ve daha yüksek arama sonuçları elde edilmesini sağlar.

HTML5 Yapısı

HTML5 ile web sayfası hazırlarken hangi bölüme ne yazılması gerektiği de oldukça önemlidir.

HTML5 kod yapısı, bir sayfada kullanılabilecek etiketleri ve bunların nasıl kullanılacağını belirtir. Bir sayfanın genel yapısını belirleyen bir 'doctype' tanımlayıcısı ile başlar. Bu tanımlayıcı, tarayıcının sayfayı nasıl render edeceğini belirtir. Örneğin:

```
<!DOCTYPE html>
```

Daha sonra, bir 'html' etiketi ile sayfanın içeriğini tanımlayan bir kök eleman (root element) oluşturulur. İçeriğin tümü bu etiketin içinde yer alır. Örneğin:


```
<html>  
...  
</html>
```

HTML5 kod yapısı, bir sayfada metin, resim, video gibi içeriği tanımlamak için etiketler kullanır. Bu etiketler, 'html' kök elemanının içinde yer alır. Örneğin, bir metni tanımlamak için 'p' etiketi kullanılır:

```
<html>  
<body>  
  <p>Bu bir metindir.</p>  
</body>  
</html>
```

Head

Head kısmı, web sayfasının meta verilerini içermektedir. Bu veriler arasında, sayfanın başlığı, anahtar kelimeleri, yazarı, stil dosyaları, açıklaması ve JavaScript dosyaları gibi önemli bilgiler yer almaktadır.



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta name="keywords" content="anahtar kelimeler" />
5     <meta name="author" content="yazar" />
6     <meta name="description" content="website hakkında bir açıklama" />
7     <link rel="stylesheet" href="style.css" />
8     <script src="script.js"></script>
9     <title>Sayfa Başlığı</title>
10  </head>
11  <body>
12    ....
13  </body>
14 </html>
15
```


Yukarıdaki örnekte, <head> etiketi içinde sayfanın başlığı, anahtar kelimeleri, yazarı ,açıklaması ve stil dosyalarını içeren meta verileri bulunmaktadır. Head kısmı, web sayfasının geri kalanından ayrılmıştır ve genellikle <body> etiketinin hemen üstünde bulunur.

Bu kısım, tarayıcının ve arama motorlarının web sayfası hakkında bilgi edinmesi için tasarlandı. Bu nedenle, head kısmı önemlidir ve düzgün bir şekilde kullanılmalıdır.

Head içi tanımlar

HTML5 sadece Doctype tanımını değil ayrıca <head> içinde kullandığımız etiketleri de ciddi manada basit hale getirmiştir.

Karakter kodu için kullandığımız

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

Yerine HTML5 karşılığı

```
<meta charset="UTF-8">
```

Gelelim script eklemek için kullandığımız koda;

```
<script type="text/javascript" src="dosya_adi.js"></script>
```

Yerine HTML5 karşılığı

```
<script src="file.js"></script>
```

Benzer şekilde ilişkilendirilmiş dokümanlar için kullandığımız

```
<link rel="stylesheet" href="style-original.css" type="text/css" />
```

Yerine HTML5 karşılığı

```
<link rel="stylesheet" href="style-original.css" />
```

Body

Body kısmı, web sayfasının kullanıcı tarafından içeriğini içermektedir. Bu içerik, metin, resimler, form elemanları, bağlantılar ve daha birçok şey olabilir.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sayfa Başlığı</title>
8 </head>
9 <body>
10  <h1> İçerik Başlığı </h1>
11  <p>Bu bir paragraf.</p>
12 </form>
13  <input type="text" placeholder="Girilmesini istediğiniz veriyi özetleyen yazı">
14  <button type="submit">Gönder</button>
15 </form>
16 </body>
17 </html>
```

Yukarıdaki örnekte, <body> etiketi içinde sayfanın başlığı, bir paragraf, bir resim, bir bağlantı ,bir form ve buton bulunmaktadır. Body kısmı, web sayfasının head kısmından ayrılmıştır ve genellikle <head> etiketinin hemen altında bulunur.

Bu kısım, web sayfasının gerçek içeriğini içermektedir ve bu içerik, kullanıcılar tarafından görüntülenir. Bu nedenle, body kısmı web sayfasının en önemli kısmıdır ve düzgün bir şekilde kullanılmalıdır.

Body etiketi içinde kullanılanlar;

HTML5 <body> etiketi içinde birçok farklı etiket kullanılabilir. Bu etiketler, web sayfasının gerçek içeriğini oluşturur ve kullanıcılar tarafından görüntülenir. Örnek olarak, aşağıdaki etiketler <body> etiketi içinde kullanılabilir:

- <h1>-<h6> etiketleri: Başlıkları belirtmek için kullanılır.
- <p> etiketi: Paragrafı belirtmek için kullanılır.
- etiketi: Resimleri belirtmek için kullanılır.
- <a> etiketi: Bağlantıları belirtmek için kullanılır.
- <form> etiketi: Form elemanlarını belirtmek için kullanılır.
- <table> etiketi: Tabloları belirtmek için kullanılır.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 <title>Sayfa Başlığı</title>
8 </head>
9 <body>
10 <h1>İçerik Başlığı</h1>
11 <p>Bu bir paragraf.</p>
12 
13 <a href="https://örnek.com"></a>
14 <form>
15 <input
16   type="text"
17   placeholder="Girilmesini istediğiniz veriyi özetleyen yazı"
18 />
19 <button type="submit">Gönder</button>
20 </form>
21 <table>
22 <tr>
23   <th>Başlık 1</th>
24   <th>Başlık 2</th>
25 </tr>
26 <tr>
27   <th>Veri 1</th>
28   <th>Veri 2</th>
29 </tr>
30 </table>
31 </body>
32 </html>
33
```

Bu örnekte, <body> etiketi içinde bir başlık, bir paragraf, bir resim, bir bağlantı, bir form ve bir tablo bulunmaktadır.

Bu etiketler, web sayfasının gerçek içeriğini oluşturur ve kullanıcılar tarafından görüntülenir.

HTML5 Content Model (İçerik Modeli)

Html5 dilinde elemanlar saklayabildikleri içerik türlerini göre sınıflandırılmıştır. Bu sınıflandırmaya İçerik Modeli (Content Model) denir. Content model elemanların ne çeşit içeriğe sahip olabileceğini tanımlar. Bir eleman birden fazla Content Modeli destekleyebilir.

Html5 tarafından belirlenen içerik türleri şunlardır;

Embedded

Dış kaynaklı, başka bir etiketleme dili ile veya programatik olarak belirlenen içeriklerdir.

Örnek: iframe, img, video, svg, object, math, embed, audio, canvas.

Flow

Gömülü olarak tanımlanabilen içeriklerdir.

Örnek: *address, article, aside, a, abbr*

Heading

Başlık tanımlamaya yarayan içerik türüdür.

Örnek: *hgroup, h1, h2, h3, h4, h5, h6*

Sectioning

Nav, section, article ve *aside* gibi elemanların içerik türüdür.

Metadata

Sayfanın görselliğini ve davranışını değiştirebilen içerik türüdür.

Örnek: *command, link, meta, base, title, style, script, noscript*.

Phrasing

Satır içi html elemanlarının ve metinlerin oluşturduğu içerik türüdür.

Örnek: *strong, small, em, a, abbr, mark*.

Interactive

Kullanıcı ile etkileşim kurmak için kullanılan interaktif içerik türüdür.

Örnek: *button, textarea, select, text, iframe, embed, label, details*

HTML5'de ID ve Class Özellikleri

ID

Bir HTML elemanın benzersiz kimliğini belirtir ve bu elemana tek bir kez kullanılabilir. Örneğin, `<div id="header">` kodu ile bir div etiketine header isimli bir id atayabiliriz ve bu div elemanına sadece header isimli bir id ile erişebiliriz. Bu sayede web sayfamızda sadece bir tane header isimli bir div elemanı bulunabilir.

Class

Bir HTML elemanına birden fazla etiket atayabileceğimiz bir anahtar olup, aynı class ismini birden fazla elemana atayarak bu elemanların ortak özelliklerini belirleyebiliriz. Örneğin, `<p class="örnek">` ve `<h1 class="örnek">` kodları ile bir paragraf ve bir başlık elemanına örnek isimli bir class atayabiliriz ve bu elemanların ortak özelliklerini article isimli class ile belirleyebiliriz.

Class ve id farkı

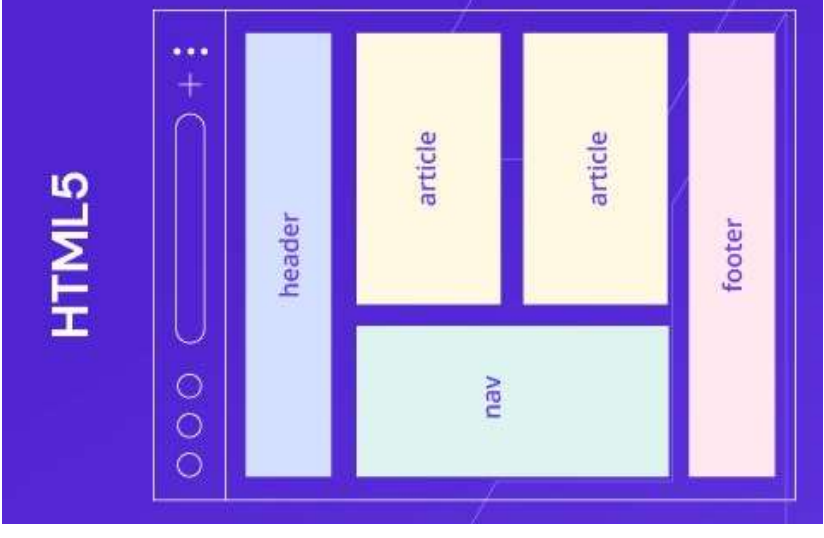
Id ve class kullanımı ve farkı arasında en önemlisi kullanım amaçlarıdır. id kullanılarak web sayfasında tek bir elemana benzersiz bir kimlik verirken, class ile birden fazla elemana ortak bir kimlik verebiliriz. id kullanımı ile bir elemana tek bir kez erişebilirken, class kullanımı ile aynı class ismini kullanan birden fazla elemana erişebiliriz.

HTML5 kod yazarken id ve class gibi özel anahtarları kullanarak CSS yardımıyla web sayfamızdaki elemanların benzersiz ve ortak özelliklerini belirleyebilir ve bu sayede web sayfalarımızı daha anlaşılır ve kullanışlı hale getirebiliriz.

Semantik HTML5 Nedir?

Semantik HTML5, HTML5 işaret dili için bir tasarım yaklaşımıdır. Bu yaklaşım, HTML etiketlerini kullanarak web sayfalarının yapısını ve içeriğini daha anlamlı bir şekilde belirtmeyi amaçlar.

Bu, arama motorlarının, web sayfalarının anlamını daha iyi anlamasına ve daha iyi bir arama sıralaması elde etmesine yardımcı olur.



Semantik HTML5, HTML etiketlerini daha doğru bir şekilde kullanarak web sayfalarının anlamını daha açık bir şekilde belirtir.

Örneğin, "header" ve "footer" etiketleri, bir sayfanın başlık ve altbilgi bölümlerini belirtmek için kullanılır. Bu, eski sürümlerdeki genel "div" etiketlerine göre daha anlamlı ve açık bir yaklaşımdır.

Web geliştiricileri için önemli bir adımdır. Bu yaklaşım sayesinde, web sayfaları daha iyi anlaşılabilir ve arama motorları tarafından daha iyi değerlendirilebilir hale gelir.

Yeni eklenen yapısal etiketler;

- **header** : Başlık eklemek için kullanabiliriz oldukça sade ve kolay şekilde HTML5 için düzenlenmiş.
- **footer** : Footer Etiketini sayfanın altındaki copyright gibi yazıları yazabileceğiniz etiket.
- **article** : Bölümlerdeki makale içeriklerini belirlemek için kullanılır
- **section** : Uygulama yada dökümanın genel bölümünü belirtir bu etiket yardımı ile aynı temayı içeren içerikleri bir arada sunabilirsiniz
- **aside** : Manşet Bloklarını belirlemek için kullanabiliriz birde sidebarlar gibi bölümler oluşturulabilir
- **nav** : Sayfada yer alan bir navigasyon bölümünü işaretlemek için kullanılır.

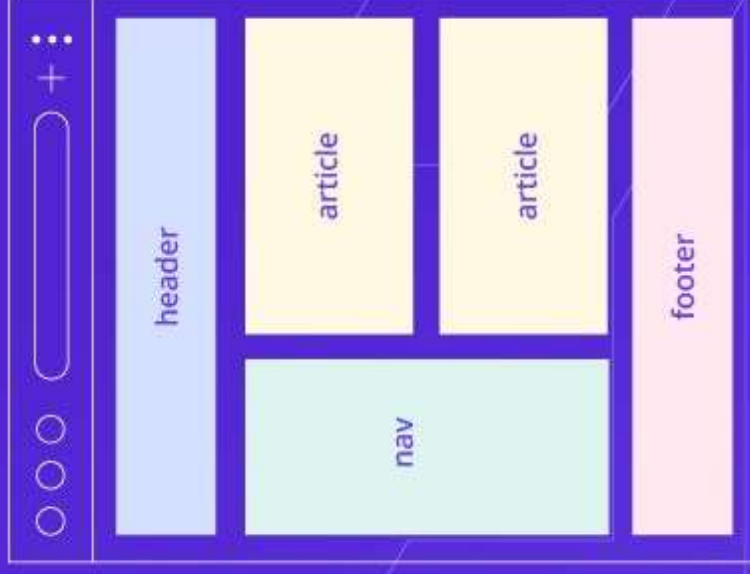
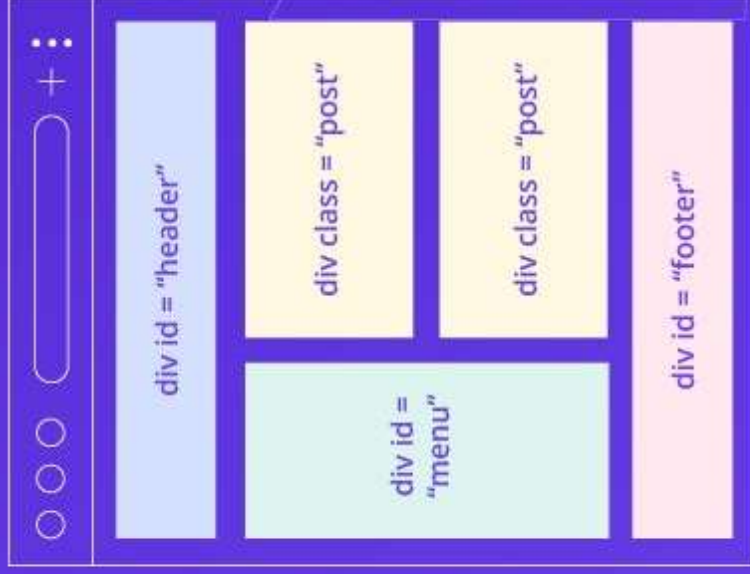
- main : Sayfada yer alan ana içeriği işaretlemek için kullanılır.
- audio : Sayfada yer alan bir audio dosyasını işaretlemek için kullanılır.
- video : Sayfada yer alan bir video dosyasını işaretlemek için kullanılır.
- canvas: Sayfada yer alan bir 2D veya 3D grafik alanını işaretlemek için kullanılır.
- dataList : Verinin sunumunu akış veya tablo biçiminde belirleyebilmeye imkan sağlar.

Bu yeni elementler, eski HTML sürümlerindeki genel "div" öğelerine göre daha anlamlı ve açık bir yaklaşım sunar.

HTML

HTML5

VS



Eklenen satır içi(inline) Elementler;

Html 4'de satır içi element olarak 5 kod var **var**, **code**, **kbd**, **tt**, **samp**. Html5 bunlara birkaç tane daha eklemiş.

- **m(mark)** : m elementi işaretlenmiş metin olarak tanımlanabilir. Anahtar kelime tabanlı çalışan arama motorları için biçilmiş kaftan olarak değerlendirilebileceğimiz bu element, programcılarında işine yarayacaktır

```
<p> Gelirken <mark>süt</mark> almayı unutma.</p>
```

- **time** : Metin içindeki zaman bilgilerini normal içerikten ayırmamıza yarayan bir

```
element.<p>Yarın<time datetime="2004-01-10 20:00">Ahmet'in doğum günü var</time>.</p>
```

=>Ekranada bir çıktısı olmaz

- **meter :** Meter elementi özel oranları tanımlayan sayısal değerleri için kullanılır. Altı adet öz niteliği vardır: value, min, low, high, max, optimum
`<label for="disk_d">Disk Kullanımı Yerel Disk D:</label>`
`<meter id="disk_d" value="6" min="0" max="10">60%</meter>`
- **progress :** İlerleyen işleğin durumunu göstermek için tanımlanır. Yüklenen resimler için kullanılabilir %45 yüklendi yazısını göstermek için mesela. İki adet öz niteliği vardır: **value** ve **max**. Javascript yardımıyla kolaylıkla yükleme işlemlerimizi yapabilmemizi sağlayacak güzel bir element.
`<label for="file">Yükleme işlemi sürüyor</label>`
`<progress id="file" value="32" max="100"> 32% </progress>`

Yeni eklenen öznitelikler(attributes);

- autofocus: Bu özellik, sayfa yüklendiğinde bir girdinin otomatik olarak odaklanmasını sağlar.

```
<input type="text" autofocus> => Sayfa yüklendiğinde kullanıcı otomatik olarak olarak odaklanır
```

- pattern: Veri giriş alanlarından, kullanıcıdan alınan verinin belirli bir formatta girilmesini sağlamak için kullanılır

```
<input type="password" pattern=".{8,}" > => Girilen şifrenin en az 8 karakter olması gerektiğini belirtir
```

- placeholder: Bir giriş alanına ipucu ekleyerek kullanıcıya o alanda ne tür bilgilerin

girilmesinin beklendiğini söyler.

```
<input type="tel" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}"> => Telefon
```

numarası inputu için placeholder ile kullanıcıya veriyi nasıl girmesi gerektiğini ve pattern ile formatı belirledik.

- required: Formun gönderilebilmesi için bir giriş alanının doldurulması gerektiğini

belirtir.

```
<input
```

```
type="text" id="username" required> => Required özelliği sayesinde Username girilmediği
```

takdirde form gönderilmez.

- **multiple**: Tanımlandığında giriş alanı için birden fazla değer girebilmeyi belirtir.

`<input type="file" multiple>` => multiple değeri girildiğinde kullanıcıya birden fazla dosya seçebilmesi için izin verilir.

- **formnovalidate**: Form verilerinin kontrol yapılmadan gönderileceğini belirtir.

`<input type="submit" formnovalidate="formnovalidate" value="Doğrulama yapmadan gönder">`

- **formtarget**: Form verilerinin hangi pencerede gönderileceğini belirtir.

`<input type="submit" formtarget="_blank|_self|_parent|_top|frameismi">`

Bunlar, HTML5'te eklenen yeni attribute lardan yalnızca birkaçı. Bu yüzden daha fazlasını öğrenmek istiyorsanız biraz daha araştırma yapmanızı öneririm.

Bazı attribute örnekleri;

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sayfa Başlığı</title>
8 </head>
9 <body>
10  <h1> İçerik Başlığı </h1>
11  <p>Bu bir paragraf.</p>
12  <form>
13    <input type="text" autofocus>
14    <button type="submit">Gönder</button>
15  </form>
16 </body>
17 </html>
```

Bu örnekte, `<input>` elemanına ``autofocus`` özelliği eklenmiştir. Bu özellik sayesinde, sayfa yüklendiğinde `<input>` elemanı otomatik olarak odaklandırılır ve kullanıcı tarafından bir değer girilebilir.

Bu özelliğin kullanımı, form doldurma işleminin hızlandırılmasına yardımcı olur.

Bu örnekte, <input> elemanına `required` özelliği eklenmiştir. Bu özellik sayesinde, <input> elemanına bir değer girilmesi zorunlu hale gelir.

Form gönderildiğinde, boş bir <input> elemanının olup olmadığı kontrol edilir ve eğer boşsa kullanıcıya bir hata mesajı gösterilir.

Ayrıca örnekte html5 ile gelen article etiketinin doğru kullanımını da görebilirsiniz

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 <title>Sayfa Başlığı</title>
8 </head>
9 <body>
10 <article>
11 <h1>İçerik Başlığı</h1>
12 <p>İlgili metin..</p>
13 </article>
14 <form>
15 <input type="text" required />
16 <button type="submit">Gönder</button>
17 </form>
18 </body>
19 </html>
20
```


Değişen etiketler;

- `<label>` tagi `<input>` etiketi için bir başlık tanımlar. `<label`
- `for="peynir">Peynir sever misin?</label>` `<input type="checkbox" />`
- `<noscript>` tarayıcısında script leri devre dışı bırakmış olan kullanıcıya bir uyarı mesajı vermek için kullanılır. `<noscript>Tarayıcınız JavaScript desteklemiyor</noscript>`
- `<s>` etiketi doğru metin tanımlamak için kullanılır. Yazının üstünü çizer. `<s>text</s>`
- `<hr>` etiketi HTML5 ile belgede bir konu sonu tanımlar. Tagi yazdığınız kısma bir çizgi çeker. `<hr>`

- `<small>` etiketi küçük metin tanımlar.
`<p><small>Bu text daha küçük</small></p>`
- `` tagi bir yazının kalın görünmesini yani bold olmasını sağlamaktadır.
`Bu yazı önemli!`
- `<u>` etiketi yazının altını çizmeye yarar.
`<p> Bu <u>yanlış</u> yazılmış bir yazıdır.</p>`
- `<address>` etiketi belge veya makalenin, yazarının/sahibinin iletişim bilgilerini tanımlar.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sayfa Başlığı</title>
8 </head>
9 <body>
10  <h1> İçerik Başlığı </h1>
11  <p>Bu bir paragraf.</p>
12  <address>
13    7134 Main Street <br>
14    Karşıyaka, İzmir 35510<br>
15    <a href="mailto:info@examp1.com">info@examp1.com</a>
16  </address>
17 </body>
18 </html>
```

Bu örnekte, <address> etiketi içinde bir adres bilgisi bulunmaktadır.

Bu adres bilgisi, satır sonu belirten
 etiketleriyle ayrılmıştır ve ayrıca bir e-posta adresine de bağlantı verilmiştir.

HTML `<noscript>` etiketi, bir web sayfasında JavaScript kodunun desteklenmediği durumlarda görüntülenecek içeriği belirtmek için kullanılır.

Ayrıca örnekte `html5` ile eklenen `aside` taginin kullanımını görebilirsiniz

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Sayfa Başlığı</title>
5 </head>
6 <body>
7   <aside>
8     <p>Yan içerik.</p>
9   </aside>
10  <script>
11    // JavaScript kodu burada bulunur
12  </script>
13  <noscript>
14    Bu sayfa JavaScript kodunu desteklemektedir. Lütfen lütfen sayfanın doğru
15    çalışabilmesi için JavaScript etkinleştiriniz
16  </noscript>
17 </body>
18 </html>
19
```

Eski etiketler;

- **<frame>**: Bir sayfanın içeriğini çerçeveler halinde bölerek her bir çerçevenin ayrı bir HTML belgesi içerisinde bulunmasını sağlar. HTML5 ile birlikte bu eleman kaldırılmıştır ve yerine **<iframe>** elemanı kullanılması önerilmektedir.
- **<frameset>**: Bir sayfanın içeriğini çerçeveler halinde bölmek için kullanılır ve **<frame>** elemanlarını içerebilir. HTML5 ile birlikte bu eleman kaldırılmıştır ve yerine **<iframe>** elemanı kullanılması önerilmektedir.
- **<basefont>**: Sayfanın varsayılan yazı tipini belirtmek için kullanılırdı. HTML5 ile birlikte bu eleman kaldırılmıştır ve yerine CSS kullanılması önerilmektedir.

- `<center>`: Sayfanın içeriğini ortalamak için kullanılırdı. HTML5 ile birlikte bu eleman kaldırılmıştır ve yerine CSS kullanılması önerilmektedir.
- `<big>`: Sayfanın içeriğindeki metni büyütmek için kullanılırdı. HTML5 ile birlikte bu eleman kaldırılmıştır ve yerine CSS kullanılması önerilmektedir.
- `<tt>`: Sayfanın içerisindeki metni teletype (düz yazı) olarak göstermek için kullanılırdı. HTML5 ile birlikte bu eleman kaldırılmıştır ve yerine CSS kullanılması önerilmektedir.

Yeni API'ler

HTML5, web sayfalarının daha gelişmiş özellikler sunabilmesi için birçok farklı API'yi de içermektedir. Bu API'ler, web sayfalarının daha dinamik ve interaktif hale gelmesini sağlar. Örnek olarak, aşağıdaki HTML5 API'leri sayılabilir:

- **Geolocation API:** Kullanıcının bulunduğu coğrafi konumu algılamak için kullanılır.
`var lokasyon = navigator.geolocation`
- **Web Workers API:** Tarayıcıda arka planda işlemler yapmak için kullanılır.
- **Web Sockets API:** Tarayıcı ve sunucu arasında gerçek zamanlı bağlantı kurmak için kullanılır
- **Canvas API:** Web sayfalarında 2 boyutlu veya 3 boyutlu grafikler oluşturmak için kullanılır

- **Html Drag and Drop Api:** İstedığınız elementi sürüklenebilir hale getirir.
`<div draggable="true">Sürüklenebilir element</div>`
- **Web Storage API:** Tarayıcıda veri saklamak için kullanılır .
 - **sessionStorage:** Veriyi geçici olarak saklar sayfa yenilendiğinde silinir.
 - **localStorage:** Veriyi kalıcı olarak saklar sayfa yenilense bile veri depoda kalır.