

Neler Yeni?

- Yeni Elementler
- Yeni Özellikler (Attributes)
- Full CSS3 Desteđi
- Video ve Ses
- 2D/3D Grafikler
- Yerel Depolama
- Yerel SQL Veritabanı
- Web Uygulamaları

HTML5 Multimedia

HTML5 ile video ve ses dosyası oynatmak her zamankinden daha kolay

- HTML5 [<video>](#)
- HTML5 [<audio>](#)

HTML5 Grafikler

HTML5 ile grafikler çizmek çok daha kolay:

- [<canvas>](#) elementini kullanmak
- Inline [SVG](#) elementini kullanmak

HTML5 Uygulamaları

HTML5 ile web uygulamaları geliřtirmek çok daha kolay.

- Yerel veri depolama
- Yerel dosya eriřimi
- Yerel SQL Veritabanı
- Uygulamaları ön belleęe alma
- Javascript çalışanları
- XMLHttpRequest 2

Semantik Elementler

Başlıklar, alt başlıklar, menüler, kısımlar ve metinler için yeni elementler.

HTML5 Formlar

Yeni form elementleri, yeni özellikler, yeni input tipleri, otomatik doęrulama.

HTML5 CSS3 kullanır

- Yeni Seçiciler
- Yeni Özellikler
- Animasyonlar
- 2D/3D Dönüşümleri
- Yuvarlandırılmış Köşeler
- Gölge Efektleri
- İndirilebilir Fontlar

Her Bölüm İçin Onlarca Örnek

Özel editörümüz ile HTML düzenleyebilir ve bir tık ile sonuçları görebilirsiniz.

Örnek

```
<!DOCTYPE HTML>
<html>
<body>

<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Tarayıcınız bu özelliği desteklemiyor.
</video>

</body>
</html>
```

[Kendin Dene »](#)

HTML5 <!DOCTYPE>

HTML dosyanızın HTML5 özelliklerinde olacağını tanımlamak için şu etiket kullanılır:

```
<!DOCTYPE html>
```

Minimum HTML5 Dökümanı

Aşağıda minimum özelliklere sahip bir HTML5 dosyası görülmektedir:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Başlık</title>
</head>

<body>
Döküman içeriği.....
</body>

</html>
```

HTML5 Semantik Elementler

◀ Önceki

Sonraki ▶

Semantik = Anlamlandırma / Anlam.

Semantik Elementler = Anlamı olan elementler.

Semantik elementler nelerdir?

Semantik bir element hem tarayıcıya hem de geliştiriciye kendisinin anlamını açıkça belirtir.

Semantik olmayan elementlere örnekler: <div> ve - İçerik hakkında bir şey söylemez.

Semantic elementlere örnekler: <form>, <table>, ve - İçeriği açıkça belirtir.

Tarayıcı Desteği

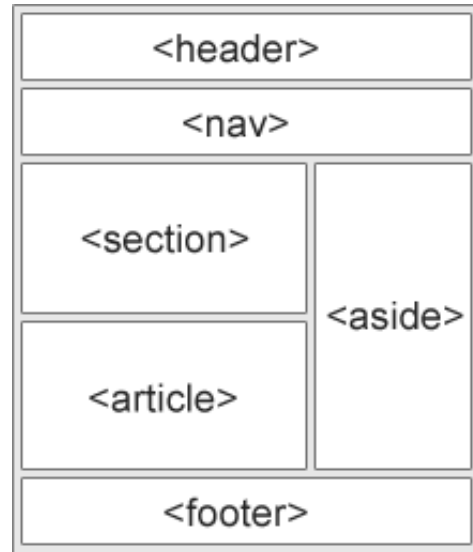


HTML5'teki Yeni Semantik Elementler

Günümüzde bir çok HTML sayfası şu şekilde kodlanmıştır: <div id="nav">, <div class="header">, or <div id="footer">,

HTML5'in bu tip bildirimler için daha iyi çözümleri vardır:

- <header>
- <nav>
- <section>
- <article>
- <aside>
- <figure>
- <figcaption>
- <footer>
- <details>
- <summary>
- <mark>
- <time>



HTML5 <section> Elementi

<section> elementi bir döküman içindeki bir kısmı belirtir.

Örnek

```
<section>
  <h1>WWF</h1>
  <p>World Wide Fund for Nature (WWF) doğanın...</p>
</section>
```

[Kendin Dene »](#)

HTML5 <article> (Makale) Elementi

Bir makale web sayfasının geri kalanından bağımsız olarak dağıtılabilmelidir.

Nerelerde kullanılabileceğine dair örnekler:

- Forum mesajları
- Blog gönderileri
- Haber Hikayesi
- Yorumlar

Örnek

```
<article>
  <h1>Internet Explorer 9</h1>
  <p>Windows Internet Explorer 9 (kısaca ....</p>
</article>
```

[Kendin Dene »](#)

HTML5 <nav> Elementi

<nav> elementi navigasyon bağlantıları büyük sayfalar için tasarlanmıştır. Ancak, belgedeki tüm linkler bu element içinde olmak zorunda değildir.

Örnek

```
<nav>
HTML Dersler
HTML Örnekler
</nav>
```

[Kendin Dene »](#)

HTML5 <aside> Elementi

<aside> elementi içindeki içerikten farklı olarak başka bazı içerikler tanımlar.

Aside içeriği, üst içerik hakkında olmalıdır.

Örnek

```
<p>Bu yaz ailemle beraber E...</p>

<aside>
  <h4>Epcot Center</h4>
  <p>Epcot Center, Florida'daki Disney World'da...</p>
</aside>
```

[Kendin Dene »](#)

HTML5 <header> Elementi

<header> elementi bir döküman ya da bir kısım (section) için bir başlık belirtir.

<header> elementi içerdiği içeriği kapsayıcı olmalıdır.

Bir dökümanda bir kereden fazla kullanabilirsiniz.

Aşağıdaki örnek bir makale (article) için bir başlık içerir:

Örnek

```
<article>
  <header>
    <h1>Internet Explorer 9</h1>
    <p><time pubdate datetime="2011-03-15"></time></p>
  </header>
  <p>Windows Internet Explorer 9 (kısaca IE9) 14.....</p>
</article>
```

[Kendin Dene »](#)

HTML5 <footer> Elementi

<footer> elementi bir döküman ya da bir kısım için alt bilgi belirtir.

Bir footer (alt bilgi) genelde dökümanın yazarını, telif haklarını, kullanım gizliliği, iletişim vs. gibi bilgileri içerir.

Bir dökümanda bir kereden fazla kullanabilirsiniz.

Örnek

```
<footer>
  <p>Gönderen: Steve Jobs</p>
  <p><time pubdate datetime="2012-03-01"></time></p>
</footer>
```

[Kendin Dene »](#)

HTML5 <figure> ve <figcaption> Elementleri

<figure> etiketi, içerisinde bir resim, gösterim, diyagram, kod listeleri vs. gibi şeylerin olduğunu belirtir.

<figure> elementi ana akış ile ilgili olsa da, konumu ana akıştan bağımsızdır ve çıkarılırsa dökümanın akışını engellemez.

<figcaption> etiketi, <figure> elementinin belirttiği resime başlık koymaya yarar..

Örnek

```
<figure>
  
  <figcaption>Fig1. - The Pulpit Pock, Norway.</figcaption>
</figure>
```

[Kendin Dene »](#)

[◀ Önceki](#)

[Sonraki ▶](#)

HTML5 Giriş (Input) Tipleri

◀ Önceki

Sonraki ▶

HTML5 Yeni Giriş Tipleri

HTML5 formlar için bir kaç yeni input tipine sahiptir. Bu yeni özellikler daha iyi kontrol ve doğrulama olanağı sağlar.

Bu bölüm şu yeni input tiplerini içerir:

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week



Tüm tarayıcılar yeni giriş tiplerini desteklemeyebilir. Ancak, yine de kullanabilirsiniz, sadece sıradan metin giriş alanı olarak davranacaklardır

Giriş Tipi: color

Giriş alanının bir renk içermesi için color kullanılır.

Örnek



Renk kodunu gösterme:

Bir renk seçin:

[Kendin Dene »](#)

Giriş Tipi: date

Bir tarih seçmenize yardımcı olur.

Örnek



Bir tarih seçme:

Doğum günü:

[Kendin Dene »](#)

Giriş Tipi: datetime

Bir tarih ve saat seçmenizi sağlar (time zone ile).

Örnek



Bir tarih ve saat seçme:

Doğum günü (tarih ve saat):

[Kendin Dene »](#)

Giriş Tipi: datetime-local

Bir tarih ve saat seçmenizi sağlar (time zone olmadan).

Örnek



Bir tarih ve saat seçme:

Doğum günü (tarih ve saat):

[Kendin Dene »](#)

Giriş Tipi: email

Input alanında bir e-posta adresi olması gerektiğinde kullanılır.

Örnek



Bir e-posta adresi girme (Form gönderildiği sırada doğrulanacaktır):

E-mail:

[Kendin Dene »](#)

İpucu: iPhone'lardaki Safari, email tipini tanır ve kendisi tamamlar (@ ekler ve .com seçeneklerini sunar)

Giriş Tipi: month

Kullanıcının bir ay ve yıl seçmesine olanak verir.

Örnek



Bir ay ve yıl seçme (time zone olmadan):

Doğum günü (ay ve yıl):

[Kendin Dene »](#)

Giriş Tipi: number

Input alanı numerik bir değer gerektirdiği zaman kullanılır.

Ayrıca hangi sayıların kabul edilebileceğini belirtebilirsiniz:

Örnek



Sayı değeri girme (Sınırlama ile):

```
Miktar girin (1 ve 5 arasında): <input type="number" name="quantity" min="1" max="5">
```

[Kendin Dene »](#)

Daha keskin sınırlamalar için aşağıdaki sayfayı inceleyin:

- [max min step value](#)

Tüm sınırlamaların olduğu bir örnek: [Kendin Dene](#)

Giriş Tipi: range

Belirli bir aralıkta olan bir sayıyı seçmenizi sağlar

Ayrıca hangi sayıların kabul edilebileceğini belirtebilirsiniz:

Örnek



Aşağıdaki örneği inceleyin:

```
<input type="range" name="points" min="1" max="10">
```

[Kendin Dene »](#)

Daha keskin sınırlamalar için aşağıdaki sayfayı inceleyin:

- [max min step value](#)

Giriş Tipi: search

Arama işleri için search tipi kullanılır. (bir search alanı tipik bir metin (text) alanı gibi davranır).

Örnek



Aşağıdaki örneği inceleyin:

```
Google'da ara: <input type="search" name="googlesearch">
```

[Kendin Dene »](#)

Giriş Tipi: tel

Örnek



Telefon numarası girmeye yarar:

```
Telefon: <input type="tel" name="usrtel">
```

[Kendin Dene »](#)

Giriş Tipi: time

Bir saat seçmenizi sağlar.

Örnek



Bir saat seçmeye yarayan örnek (time zone olmadan):

Bir saat seçin:

[Kendin Dene »](#)

Giriş Tipi: url

Bir giriş alanı bir URL bağlantısı içerecekse kullanılır.

Bu özellik, form gönderildiği sırada doğrulanır.

Örnek



Bir adres girme:

Ana sayfanızı ekleyin:

[Kendin Dene »](#)

Giriş Tipi: week

Bir hafta ve yıl seçmenizi sağlar.

Örnek



Bir hafta ve yıl seçme (time zone olmadan):

Bir hafta seçin:

[Kendin Dene »](#)

HTML5 Form Elementleri

◀ Önceki

Sonraki ▶

HTML5 Yeni Form Elementleri

HTML5 aşağıdaki yeni form elementlerine sahiptir:

- <datalist>
- <keygen>
- <output>



Tüm tarayıcılar yeni form elementlerini desteklemeyebilir. Ancak, yine de kullanabilirsiniz, sadece sıradan metin giriş alanı olarak davranacaklardır.

HTML5 <datalist> Elementi

<datalist> elementi, bir <input> elementi için önceden belirlenmiş bir dizi seçeneğin listesini belirtir.

<datalist> elementi <input> elementleri için bir "autocomplete" özelliği sağlar. Kullanıcılar, input verisi için önceden tanımlanmış bir açılır kutu listesi göreceklerdir.

Bir <datalist> elemanı ile birlikte bağlamak için <input> elemanın "list" özelliğini kullanın.

Örnek



Bir <datalist> ile önceden değerleri belirlenmiş bir <input> elementi:

```
<input list="browsers">

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

[Kendin Dene »](#)

HTML5 <keygen> Elementi

<keygen> elementinin amacı, kullanıcıları yetkilendirmek için güvenli bir yol kullanmaktır.

<keygen> etiketi, bir form içerisinde bir anahtar çifti (key-pair) jeneratörü belirtir.

Form gönderildiğinde, iki anahtar oluşturulur, biri özel biri ise genel.

Özel anahtar sizin lokalinizde saklanır, genel olanı ise sunucuya gönderilir. Genel anahtar, kullanıcının ileride tekrar kullanabilmesi için bir istemci sertifikası oluşturmak için kullanılır.

Örnek



Keygen alanı olan bir form:

```
<form action="demo_keygen.php" method="get">
Kullanıcı adı: <input type="text" name="usr_name">
Şifreleme: <keygen name="security">
<input type="submit">
```

</form>

[Kendin Dene »](#)

HTML5 <output> Elementi

<output> elementi bir hesaplamamın sonucunu gösterir (bir script gibi)

Örnek



Bir hesaplama yapmak ve sonucu <output> elementi içinde göstermek:

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">0
<input type="range" id="a" value="50">100 +
<input type="number" id="b" value="50">=
<output name="x" for="a b"></output>
</form>
```

[Kendin Dene »](#)

HTML5 Form Özellikleri

◀ Önceki

Sonraki ▶

HTML5 Yeni Form Özellikleri

HTML5 <form> ve <input> için bir kaç yeni özelliğe sahiptir.

<form> için yeni özellikleri:

- autocomplete
- novalidate

<input> için yeni özellikler:

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height ve width
- list
- min ve max
- multiple
- pattern (regexp)
- placeholder
- required
- step

<form> / <input> autocomplete özelliği

autocomplete özelliği bir form ya da input alanının autocomplete özelliğinin on ya da off olacağını belirtir.

autocomplete açık olduğunda, tarayıcı otomatik olarak kullanıcı tarafından önceden girilen bilgileri tamamlar.

İpucu: form'larda autocomplete on yapmak ve belli input alanlarında off yapmak mümkündür, ya da tam tersi.

Not: autocomplete özelliği <form> ile ve şu <input> tipleri ile çalışır: text, search, url, tel, email, password, datepickers, range, ve color.

Örnek



Autocomplete on olan bir HTML form'u. (ve input alanı için off):

```
<form action="demo_form.php" autocomplete="on">
  İsim:<input type="text" name="fname"><br>
  Soyisim: <input type="text" name="lname"><br>
  E-mail: <input type="email" name="email" autocomplete="off"><br>
  <input type="submit">
</form>
```

Kendin Dene »

İpucu: Bazı tarayıcılarda bu örneğin çalışması için "otomatik tamamlama" özelliğini aktifleştirmeniz gerekebilir.

<form> novalidate Özelliği

novalidate özelliği bir 0/1 özelliğidir (boolean) (TRUE - FALSE).

Kullanıldığında, form verisinin (input) doğrulanmasına gerek olmadığını belirtir.

Örnek



Gönderileceği zaman, formun doğrulanmadığını belirtir:

```
<form action="demo_form.php" novalidate>
  E-mail: <input type="email" name="user_email">
  <input type="submit">
</form>
```

[Kendin Dene »](#)

<input> autofocus Özelliği

Autofocus özelliği bir 0/1 özelliğidir (boolean) (TRUE - FALSE).

Kullanıldığında, sayfa yüklendiği anda bir <input> elementine odaklanılmasını sağlar. Yani, ilk başta o kısma veri girmenizi önerir.

Örnek



Sayfa yüklendiğinde "İsim" kısmına odaklanmayı mümkün hale getirelim:

```
İsim:<input type="text" name="fname" autofocus>
```

[Kendin Dene »](#)

<input> form Özelliği

Form özelliği, bir <input> elementinin hangi form ya da formlara ait olduğunu belirtir.

İpucu: Bir form'dan fazla ise form'ların id'lerini aralarında boşluklar bırakarak giriniz.

Örnek



HTML form elementinin dışında olmasına rağmen, halen o form'a ait olan input alanı:

```
<form action="demo_form.php" id="form1">
  İsim: <input type="text" name="fname"><br>
  <input type="submit" value="Submit">
</form>
```

```
Soyisim: <input type="text" name="lname" form="form1">
```

[Kendin Dene »](#)

<input> formaction Özelliği

Formaction özelliği, bir form gönderildiğinde giriş denetimi (input control) işleyecek bir dosyanın URL'sini belirtir.

Formaction <form> elemanın eylem özniteliği geçersiz kılar.

Not: Formaction özelliği type="submit" ve type="image" ile kullanılır.

Örnek



Farklı iş gören iki submit (gönder) butonu olan bir HTML form:

```
<form action="demo_form.php">
  İsim: <input type="text" name="fname"><br>
  Soyisim: <input type="text" name="lname"><br>
  <input type="submit" value="Submit"><br>
  <input type="submit" formaction="demo_admin.php"
  value="Submit as admin">
</form>
```

[Kendin Dene »](#)

<input> formenctype Özelliği

Formenctype özelliği, form-data'nın sunucuya gönderilirken nasıl şifrelenmesi gerektiğini belirtir. (sadece method="post" olan form'lar için)

Formenctype özelliği, <form> elemanının enctype özniteliği geçersiz kılar.

Not: Formenctype özelliği type="submit" ve type="image" ile kullanılır.

Örnek



Birinci submit butonu normal kodlanırken, ikinci olan ise "multipart/form-data" olarak kodlanmıştır:

```
<form action="demo_post enctype.php" method="post">
  İsim: <input type="text" name="fname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formenctype="multipart/form-data"
  value="Submit as Multipart/form-data">
</form>
```

[Kendin Dene »](#)

<input> formmethod Özelliği

Formmethod özelliği, eylem URL'sine form-data göndermek için HTTP yöntemini tanımlar

Formmethod özelliği, <form> elementinin method özelliğini geçersiz kılar.

Not: Formmethod özelliği type="submit" veytype="image" ile kullanılabilir.

Örnek



İkinci gönderme düğmesi formun HTTP yöntemini geçersiz kılar:

```
<form action="demo_form.php" method="get">
  İsim: <input type="text" name="fname"><br>
  Soyisim: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formmethod="post" formaction="demo_post.php"
  value="Submit using POST">
</form>
```

[Kendin Dene »](#)

<input> formnovalidate Özelliği

Formnovalidate bir 0/1 özelliğidir (boolean) (TRUE - FALSE).

Kullanıldığında, <input> elementinin doğrulanmaması gerektiğini belirtir.

Formnovalidate özelliği, <form> elementinin novalidate özelliğini geçersiz kılar.

Not: Formnovalidate özelliği type="submit" ile kullanılabilir.

Örnek



İki submit butonu olan bir form (doğrulamalı ve doğrulamasız)

```
<form action="demo_form.php">
  E-mail: <input type="email" name="userid"><br>
  <input type="submit" value="Submit"><br>
  <input type="submit" formnovalidate value="Submit without validation">
</form>
```

[Kendin Dene »](#)

<input> formtarget Özelliği

Formtarget özelliği, form gönderildikten sonra alınan cevabın nerede görüntüleneceğini gösteren bir isim (name) ya da anahtar kelime (keyword) belirtir.

Formtarget özelliği, <form> elementinin target özelliğini geçersiz kılar.

Not: Formtarget özelliği type="submit" ve type="image" ile kullanılabilir.

Örnek



İki submit butonu olan ve farklı hedef pencereleri olan bir form:

```
<form action="demo_form.php">
  İsim: <input type="text" name="fname"><br>
  Soyisim: <input type="text" name="lname"><br>
  <input type="submit" value="Submit as normal">
  <input type="submit" formtarget="_blank"
  value="Submit to a new window">
</form>
```

[Kendin Dene »](#)

<input> height (yükseklik) ve width (genişlik) Özelliği

Height ve width özelliği, bir <input> elementinin yükseklik ve genişliğini belirtir.

Not: Height ve width özelliği sadece <input type="image"> ile kullanılabilir.

İpucu: Resimler için her zaman bu iki özelliği de kullanın. Yükseklik ve genişlik ayarlı ise sayfa yüklendiğinde, görüntü için gerekli alan ayrılmış olacaktır. Ancak, bu özellikler olmadan, tarayıcı görüntünün boyutunu bilmediği için buna uygun bir alan ayıramaz. Etkisi (görüntüler yüklenirken) sayfa düzeni yüklemesi sırasında değişecektir.

Örnek



Submit butonu olarak yüksekliği ve genişliği ayarlanmış olan bir resim tanımlayın:

```
<input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
```

[Kendin Dene »](#)

<input> list Özelliği

List özelliği, bir <input> elemanı için önceden tanımlanmış seçenekler içeren bir <datalist> element anlamına gelir.

Örnek



Bir <datalist> içinde değerleri önceden tanımlanmış bir <input> elementi:

```
<input list="browsers">

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

[Kendin Dene »](#)

<input> min ve max Özelliği

Min ve max özelliği, bir <input> elementi için minimum ve maksimum değerleri belirtir.

Not: Min ve max şu input tipleri ile çalışır: number, range, date, datetime, datetime-local, month, time ve week.

Örnek



Min ve max değerleri olan <input> elementleri:

```
01-01-1980'den önce bi tarih girin:
<input type="date" name="bday" max="1979-12-31">

01-01-2000'den sonra bir tarih girin:
<input type="date" name="bday" min="2000-01-02">

1 ile 5 arasında bir değer girin:
<input type="number" name="quantity" min="1" max="5">
```

[Kendin Dene »](#)

<input> multiple Özelliği

Multiple özelliği bir 0/1 özelliğidir (boolean) (TRUE - FALSE).

Kullanıldığında, kullanıcının <input> elementine 1'den fazla değer girmesine izin verir.

Not: Multiple özelliği, şu input tipleri ile çalışır: email ve file.

Örnek



1'den fazla değer kabul eden dosya yükleme alanı:

```
Resimleri seçiniz: <input type="file" name="img" multiple>
```

[Kendin Dene »](#)

<input> pattern Özelliği

Pattern özelliği, bir <input> alanına girilecek olan değer hangi düzende olması gerektiğini belirtir.

Not: Pattern özelliği şu input tipleri ile çalışır: text, search, url, tel, email ve password.

Örnek



Sadece üç harfe izin veren bir input alanı (rakam yok, özel karakter yok):

```
Ülke kodu: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Üç karakterli ülke kodu">
```

[Kendin Dene »](#)

<input> placeholder Özelliği

Placeholder özelliği, bir giriş alanının beklenen değerini açıklayan kısa bir ipucu belirtir.

Kullanıcı bir değer girmeden önce kısa bir ipucu gözükecektir.

Not: Placeholder özelliği şu input tipleri ile çalışır: text, search, url, tel, email ve password.

Örnek



Placeholder metni olan bir input alanı:

```
<input type="text" name="fname" placeholder="İsim">
```

[Kendin Dene »](#)

<input> required Özelliği

Required özelliği bir 0/1 özelliğidir (boolean) (TRUE - FALSE).

Kullanıldığında, form gönderilmeden önce istenen input alanlarının **mutlaka** doldurulması gerektiğini belirtir.

Not: Required özelliği şu input tipleri ile çalışır: text, search, url, tel, email, password, date pickers, number, checkbox, radio ve file.

Örnek



Required (zorunlu) olan bir input alanı:

```
Kullanıcı adı: <input type="text" name="username" required>
```

[Kendin Dene »](#)

<input> step Özelliđi

Step özelliđi, bir input alanı için girilebilecek sayı deđerini aralıđını belirtir.

Örnek: Eğer step="3" ise, deđerler -3, 0, 3, 6, vs. olabilir.

İpucu: Bir gerçek deđerler aralıđı oluşturmak için step özelliđi max and min özellikleri ile kullanılabilir.

Not: Step özelliđi şu input tipleri ile çalışır: number, range, date, datetime, datetime-local, month, time ve week.

Örnek



Deđer aralıđı belirtilmiş bir input alanı:

```
<input type="number" name="points" step="3">
```

[Kendin Dene »](#)

HTML5 Canvas

◀ Önceki

Sonraki ▶

<canvas> elementi bir web sayfasında grafik çizmek için kullanılır.

Sol taraftaki örnek, bir tuvalin (canvas) üstünde bir kırmızı dikdörtgen, bir eğimli dikdörtgen, bir çok renkli dikdörtgen ve bir çok renkli metin göstermektedir.

Canvas (Tuval) Nedir?

HTML5 <canvas> elementi script'ler (genellikle JavaScript) yoluyla grafikler çizmek için kullanılır.

<canvas> grafikler için sadece bir tutucudur (tuval). Gerçekten bir grafik çizmek için script kullanmalısınız.

Canvas yollar, kutular, daireler, yazılar ve resimler eklemek için bir kaç değişik metod kullanır.

Tarayıcı Desteği



Internet Explorer 9+, Firefox, Opera, Chrome, ve Safari <canvas> elementin destekler.

Not: Internet Explorer 8 ve öncesi desteklemez.

Canvas oluşturmak

Bir canvas bir HTML sayfasındaki dikdörtgen bir alandır ve <canvas> elementi ile belirtilir.

Not: Varsayılan olarak <canvas> elementinin sınırları ve içeriği yoktur.

Biçimlendirme (markup) şu şekildedir:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Not: Her zaman bir "ID" etiketi (script içinde kullanmak için) ve canvas'ın boyutlarını belirtmek için bir genişlik (width) ve yükseklik (height) özelliği kullanın.

İpucu: Bir HTML sayfasında birden fazla <canvas> elementi kullanabilirsiniz.

Bir sınır kalınlığı (border) eklemek için stil (style) kullanın:

Örnek

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;"></canvas>
```

[Kendin Dene »](#)

JavaScript ile Canvas (Tuval) Üzerine Çizim Yapmak

Canvas üzerindeki tüm çizimler JavaScript içinde yapılmalıdır:

Örnek

```
<script>
```

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
</script>
```

[Kendin Dene »](#)

Örneğin açıklaması:

İlk olarak, <canvas> elementini bul:

```
var c = document.getElementById("myCanvas");
```

Sonra, getContext() metodunu çağırın. ((getContext() metodunu çağırarak için "2d" string ifadesini yazmak zorundasınız):

```
var ctx = c.getContext("2d");
```

getContext("2d") nesnesi, HTML nesnelere içinde olan, yollar, kutular, daireler, yazılar, resimler ve daha fazlasını çizmek için bir çok özelliği ve metodu olan bir gömme'dir. (built-in).

Sonraki iki satır kırmızı bir dikdörtgen çizer:

```
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
```

fillStyle özelliği bir CSS rengi, bir eğim (gradient) ya da bir desen olabilir. Varsayılan değeri ise #000000, yani siyahtır.

fillRect(x,y,width,height) metodu, o anki fillStyle ile bir dikdörtgen çizer.

Canvas Koordinatları

Canvas iki boyutlu bir ızgaradır (grid).

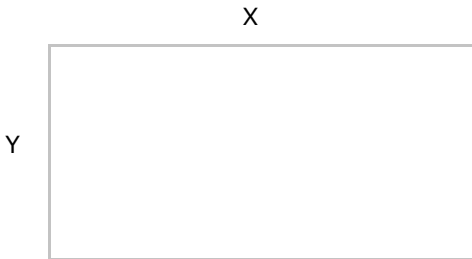
Canvas'ın sol üst koordinatları(0,0)'dir.

Ve yukarıdaki fillRect() metodu(0,0,150,75) parametlerine sahipse

Bunun anlamı: "Sol-üst köşeden başla ve 150x75 piksel boyutlarında bir dikdörtgen çiz" demektir.

Koordinat Örneği

Farenizi aşağıdaki dikdörtgenin üzerinde gezdirin ve (x,y) koordinatlarını görün. (x = yatay, y = dikey):



Canvas - Yollar (Paths)

Bir Canvas'ın üzerine düz çizgiler çizmek için, aşağıdaki metodları kullanacağız:

- moveTo(x,y) Satırın başlama noktasını belirtir
- lineTo(x,y) Satırın bitiş noktasını belirtir

Gerçekten çizgi çizmek için, "ink" metodlarından birini, örneğin stroke() komutunu kullanmak zorundayız.

Örnek

Başlangıç noktası olarak (0,0) ve bitiş noktası olarak (200,100) kullanın ve stroke() komutunu kullanarak çizgi çizin:

JavaScript:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);
ctx.lineTo(200,100);
ctx.stroke();
```

[Kendin Dene »](#)

Daire yada çember çizmek için aşağıdaki metodu kullanacağız:

- `arc(x,y,r,start,stop)`

Gerçekten bir çember çizmek için, "ink" metodlarından `stroke()` yada `fill()` komutlarını kullanmak zorundayız.

Örnek

`arc()` metodunu kullanarak çember çizmek

JavaScript:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
```

[Kendin Dene »](#)

Canvas - Metin

Canvas üzerine bir metin çizmek için en önemli özellik ve metodlar şunlardır:

- `font` - Metnin font özelliklerini belirtir.
- `fillText(text,x,y)` - Dolgulu metin çizer
- `strokeText(text,x,y)` - Metin çizer (dolgu yok)

`FillText()` Kullanımı:

Örnek

30 piksel yüksekliğinde, Arial fontunu kullanarak dolgulu bir metin yazın:

JavaScript:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.fillText("Merhaba Dünya",10,50);
```

[Kendin Dene »](#)

`strokeText()` Kullanımı:

Örnek

30 piksel yüksekliğinde, Arial fontunu kullanarak dolgunuz bir metin yazın:

JavaScript:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.font = "30px Arial";
ctx.strokeText("Merhaba Dünya", 10, 50);
```

[Kendin Dene »](#)

Canvas - Eğimler (Gradients)

Eğimler dikdörtgenleri, çemberleri, çizgileri, metinleri ya da başka şeyleri doldurmak için kullanılır. Canvas üzerindeki şekiller koyu renklerle sınırlanmaz.

There are two different types of gradients:

- `createLinearGradient(x,y,x1,y1)` - Düzgün eğim yaratır.
- `createRadialGradient(x,y,r,x1,y1,r1)` - Çembersel eğim yaratır.

Eğer bir eğimsel nesnemiz varsa, iki ya da daha fazla "color stops" eklemeliyiz.

`addColorStop()` metodu "color stops"ları ve onların gradient boyunca olan pozisyonlarını belirtir. Gradient'lerin pozisyonları 0 ve 1 arasında herhangi bir yerde olabilir.

Gradient kullanmak için, `fillStyle` veya `strokeStyle` özelliğini gradient üzerinde kullanın ve dikdörtgen, metin ya da çizgi gibi şekiller çizin.

`createLinearGradient()` Kullanımı:

Örnek

Düzgün doğrusal bir eğim yaratın. Dikdörtgeni eğimli olarak dolgulandırın:

JavaScript:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Gradient'i oluşturma
var grd = ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");

// Eğimli olarak dolgulandırma
ctx.fillStyle = grd;
ctx.fillRect(10,10,150,80);
```

[Kendin Dene »](#)

`createRadialGradient()` Kullanımı:

Örnek

Bir çembersel gradient oluşturun ve dikdörtgeni eğimli olarak dolgulandırın:

JavaScript:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");

// Gradient'i oluřturma
var grd = ctx.createRadialGradient(75,50,5,90,60,100);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");

// Eđimli olarak dolgulandırma
ctx.fillStyle = grd;
ctx.fillRect(10,10,150,80);
```

[Kendin Dene »](#)

Canvas - Resimler

Bir canvas üzerine bir resim çizmek için ařađıdaki metodu kullanacađız:

- `drawImage(image,x,y)`

Resim Kullanımı:



Örnek

Canvas üzerine resim çizmek:

JavaScript:

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
var img = document.getElementById("scream");
ctx.drawImage(img,10,10);
```

[Kendin Dene »](#)

HTML5 Inline (Sıralı) SVG

◀ Önceki

Sonraki ▶

HTML5 sıralı SVG'yi destekler.



SVG Nedir?

- SVG Skala edilebilir Vektör Grafikleri (Scalable Vector Graphics) anlamına gelir.
- SVG vektör temelli grafikleri ifade etmek için kullanılır.
- SVG grafikleri XML formatında tanımlar
- SVG grafikleri zoom'lansalar da yeniden boyutlandırılırsalar da kaliteleri düşmez.
- SVG dosyaları içindeki her element, her özellik (attribute) hareketlendirilebilir (anime)

SVG'nin Avantajları

Resim dosyalarında SVG kullanmanın avantajları (JPEG ve GIF gibi):

- SVG resimleri herhangi bir metin editörü ile yaratılabilir ya da düzenlenebilir
- SVG resimleri aranabilir, indekslenebilir, script kodu ile yazılabilir ya da sıkıştırılabilir
- SVG resimleri skale edilebilir.
- SVG resimleri tüm çözünürlüklerde yüksek kalite ile yazdırılabilir (yazıcı)
- SVG resimleri zoom edilebilir (ve kalite düşmez)

Tarayıcı Desteği



Internet Explorer 9+, Firefox, Opera, Chrome, ve Safari inline SVG'yi destekler.

SVG'yi Doğrudan HTML Sayfalarına Gömmek

HTML5 ile SVG elementlerini doğrudan sayfanıza gömebilirsiniz:

Örnek

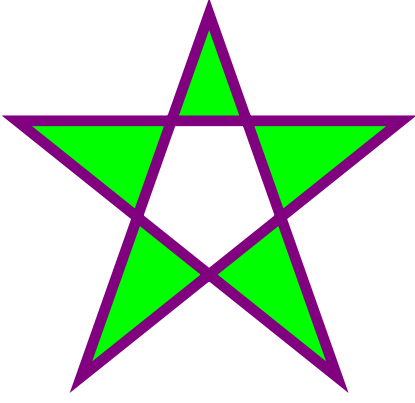
```
<!DOCTYPE html>
<html>
<body>

<svg width="300" height="200">
  <polygon points="100,10 40,180 190,60 10,60 160,180"
    style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>

</body>
</html>
```

[Kendin Dene »](#)

Sonuç:



Canvas ve SVG Karşılaştırması

Aşağıdaki tablo Canvas ve SVG arasındaki önemli farkları gösterir:

Canvas	SVG
<ul style="list-style-type: none">• Çözünürlüğe bağımlı• Olay (event) işleyicilerini desteklemez• Zayıf render kapasitesi (metinler için)• Sonuçları .png veya .jpg olarak kaydedebilme• Oyunlar için uygun	<ul style="list-style-type: none">• Çözünürlükten bağımsız• Olay (event) işleyicilerini destekler• geniş render alanları olan uygulamalar için en uygun çözüm(Google Maps)• Karmaşıkta yavaş render işlemi• Oyunlar için uygun değil

HTML5 Video

◀ Önceki Sonraki ▶

Bir çok modern web sitesi video oynatır. HTML5 bunlar için bir standar sunar.

Tarayıcınızın destekleyip desteklemediğini kontrol edin.

Kontrol Et

Tarayıcı Desteği



Internet Explorer 9+, Firefox, Opera, Chrome ve Safari destekler.

HTML5 Video - Nasıl Çalışır

HTML5'te video görüntülemek için yapmanız gereken tek şey:

Örnek

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogv" type="video/ogg">
  Tarayıcınız bu özelliği desteklemiyor.
</video>
```

Kendin Dene »

"control" özelliği, "play, pause, ses" gibi denetimleri ekler.

Ayrıca genişlik ve yükseklik ayarlarını belirtmek iyi bir fikirdir. Eğer ayarlanırsa, video için gerekli boşluk sayfa yüklenirken ayarlanır. Ancak, bunlar yoksa, tarayıcı ne kadar alan ayıracağını bilemez.

<video> elementi çoklu kaynak (<source>) kullanımına izin verir. <source> elementi farklı ses dosyalarına bağlantıya izin verir. Tarayıcı, ilk tanımlayabildiği dosyayı kullanır.

Video Formatları ve Tarayıcı Desteği

Şu an için <video> elementini destekleyen üç format bulunmaktadır: MP4, WebM ve Ogg:

Tarayıcı	MP4	WebM	Ogg
Internet Explorer	EVET	HAYIR	HAYIR
Chrome	EVET	EVET	EVET
Firefox	NO	EVET	EVET
Safari	EVET	HAYIR	HAYIR
Opera	HAYIR	EVET	EVET

- MP4 = MPEG 4 dosyaları. H264 video codec ve AAC audio codec ile
- WebM = WebM dosyaları. VP8 video codec ve Vorbis audio codec ile
- Ogg = Ogg dosyaları. Theora video codec ve Vorbis audio codec ile

Video Formatları için MIME Tipleri

Format	MIME Tipi
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML5 <video> - DOM Metodu ve Özellikleri

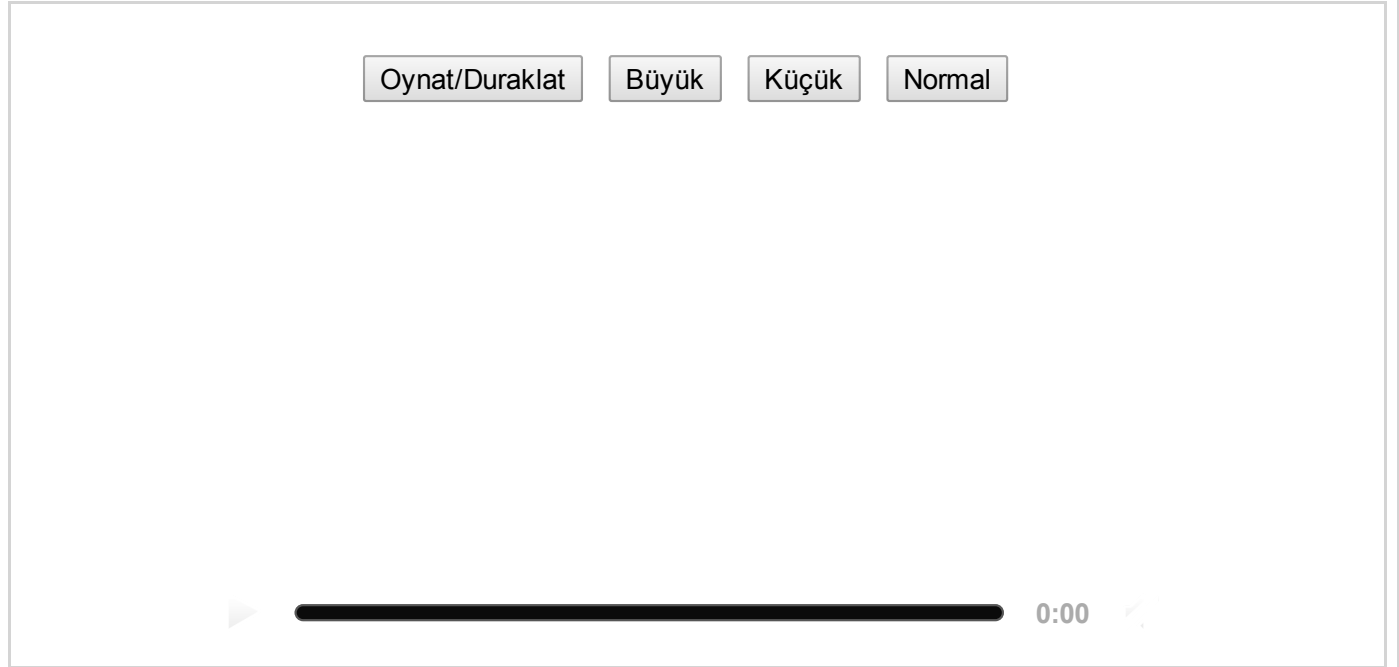
HTML5 <video> ve <audio> elementleri için DOM metodları, özellikleri ve olayları vardır.

Bu metodlar, özellikler ve olaylar JavaScript kullanarak medya dosyalarını kontrol etmenizi sağlar.

Aşağıda bunun bir örneği görülmektedir.

Örnek

Bir video için basit oynat/duraklat ve boyutlandır kontrolleri oluşturma:



Yukarıdaki örnek iki metod kullanır: play() ve pause(). Ayrıca iki özellik (property) kullanır: paused ve width.

[Kendin Dene »](#)

HTML5 Audio

◀ Önceki Sonraki ▶

HTML5 ses dosyalarının çalınması için bir standard sağlar.

Web'de Audio (Ses)

HTML5'ten önce, ses dosyalarının web'de çalınması için bir standard yoktu.

HTML5'ten önce, audio dosyalarının çalınması için eklentilere (flash gibi) ihtiyaç vardı. Bununla beraber, farklı tarayıcılar farklı eklentileri de destekliyordu.

HTML5 bir ses dosyasını bir web sayfasına gömmek için yeni bir element kullanır: <audio> element.

Tarayıcı Desteği



Internet Explorer 9+, Firefox, Opera, Chrome ve Safari <audio> elementini destekler.

Not: Internet Explorer 8 ve öncesi desteklemez.

HTML5 Audio - Nasıl Çalışır?

Bir ses dosyasını bir HTML5 dosyası içinde çalmak için yapmanız gereken tek şey aşağıdaki gibidir:

Örnek

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Tarayıcınız audio elementini desteklemiyor.
</audio>
```

[Kendin Dene »](#)

"control" özelliği, "play, pause, ses" gibi denetimleri ekler.

Ayrıca <audio> ve </audio> etiketleri arasında desteklemeyen tarayıcı uyarısı eklemek zorundasınız.

<audio> elementi çoklu kaynak (<source>) kullanımına izin verir. <source> elementi farklı ses dosyalarına bağlantıya izin verir. Tarayıcı, ilk tanımlayabildiği dosyayı kullanır.

Ses Formatları ve Tarayıcı Destekleri

Şu an için <audio> elementinin desteklediği sadece üç format vardır: MP3, Wav ve Ogg:

Tarayıcı	MP3	Wav	Ogg
Internet Explorer	EVET	HAYIR	HAYIR
Chrome	EVET	EVET	EVET
Firefox	HAYIR Update: Firefox 21 running on Windows 7, Windows 8, Windows Vista, and Android now supports MP3	EVET	EVET
Safari	EVET	EVET	HAYIR

Opera	HAYIR	EVET	EVET
-------	-------	------	------

Ses formatları için MIME tipleri

Format	MIME-type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

HTML5 Geolocation (Coğrafi Konum)

◀ Önceki

Sonraki ▶

Uyarı:

Bu özelliği test edebilmeniz için bilgisayarınızda GPS cihazı olmak zorundadır

Eğer bilgisayarınızda GPS cihazı yoksa, bu sayfayı akıllı telefonlarınızdan da açıp test edebilirsiniz

HTML5 Geolocation bir kullanıcının konumunu belirtir.

Dene

Konum Öğrenmek

Bu özellik kullanıcı gizliliğine girdiği için, kullanıcı izin vermeyene kadar çalışmaz.

Tarayıcı Desteği



Internet Explorer 9+, Firefox, Chrome, Safari ve Opera bu özelliği destekler.

HTML5 - Geolocation Kullanımı

Konumunuzu görmek için `GetCurrentPosition()` metodunu kullanınız.

Aşağıdaki örnek, kullanıcının enlem ve boylam bilgilerini göstermeye yarar:

Örnek

```
<script>
var x = document.getElementById("demo");
function getLocation()
{
  if (navigator.geolocation)
  {
    navigator.geolocation.getCurrentPosition(showPosition);
  }
  else{x.innerHTML = "Tarayıcınız bu özelliği desteklemiyor.";}
}
function showPosition(position)
{
  x.innerHTML = "Enlem: " + position.coords.latitude +
  "<br>Boylam: " + position.coords.longitude;
}
</script>
```

Kendin Dene »

Örneğin açıklaması:

- Geolocation özelliğini desteklenip desteklenmediğini kontrol et.
- Eğer destekleniyorsa, `getCurrentPosition()` metodunu çalıştır. Desteklenmiyorsa, kullanıcıya bir mesaj göster.
- Eğer `getCurrentPosition()` metodu başarılı olarak çalıştıysa, koordinatları alır. (`showPosition`)
- `showPosition()` fonksiyonu enlem ve boylam bilgilerini gösterir.

Yukarıdaki örnek, hataları göz önüne almadan çalışan çok basit bir örnektir.

Hataları ve Reddetmeleri Göz Önüne Almak

getCurrentPosition() metodunun ikinci parametresi hataları kavramak için kullanılmıştır. Eğer bir kullanıcının konumunu göstermek başarısız olursa bir çalıştırmak için bir fonksiyon belirtir:

Örnek

```
function showError(error)
{
  switch(error.code)
  {
    case error.PERMISSION_DENIED:
      x.innerHTML = "Geolocation kullanıcı tarafından engellendi."
      break;
    case error.POSITION_UNAVAILABLE:
      x.innerHTML = "Konum bilgisi bulunamadı."
      break;
    case error.TIMEOUT:
      x.innerHTML = "İstek zaman aşımına uğradı."
      break;
    case error.UNKNOWN_ERROR:
      x.innerHTML = "Bilinmeyen bir hata oluştu."
      break;
  }
}
```

[Kendin Dene »](#)

Sonuçları Haritada Göstermek

Sonuçları bir haritada göstermek için Google Maps gibi enlem ve boylam bilgilerini kullanabilen bir harita servisine ihtiyacınız vardır:

Örnek

```
function showPosition(position)
{
  var latlon = position.coords.latitude + "," + position.coords.longitude;

  var img_url = "http://maps.googleapis.com/maps/api/staticmap?center="
  +latlon+"&zoom=14&size=400x300&sensor=false";

  document.getElementById("mapholder").innerHTML = "<img src='"+img_url+"'>";
}
```

[Kendin Dene »](#)

Yukarıdaki örnekte, konumu Google Haritalar'da görüntülemek için enlem ve boylam bilgilerini kullandık.

[Google Map Script](#)

İnteraktif bir işaretleyicisi olan, zoom ve sürükleme seçenekleri olan bir harita görüntülemek için bir script nasıl kullanılır.

Konum - Spesifik Bilgi

Bu sayfa bir kullanıcının konumunu bir harita üzerinde göstermek için kullanılmıştır. Ancak, Geolocation ayrıca "Konum - Spesifik Bilgi" için de çok kullanışlıdır.

Örnekler:

- Güncel yerel bilgiler
- Kullanıcıya ilgi çekici noktalar gösterilebilir.
- Navigasyon (GPS)

getCurrentPosition() Metodu - Sonuç Verisi

getCurrentPosition() metodu başarılı çalışırsa size bir nesne dönderir. Enlem, boylam ve doğruluk özellikleri daima size bildirilir. Aşağıdaki diğer özellikler, eğer uygunlarsa size bildirilir.

Özellik	Açıklama
coords.latitude	Onluk tabanda enlem bilgisi
coords.longitude	Onluk tabanda boylam bilgisi
coords.accuracy	Pozisyon doğruluğu
coords.altitude	Deniz seviyesine göre yükseklik
coords.altitudeAccuracy	Pozisyonun yükseklik doğruluğu
coords.heading	Kuzey'den saat yönünde derece olarak rota
coords.speed	Saniyedeki hız değeri (metre olarak)
timestamp	Cevabın tarih/zaman bilgisi

Geolocation nesnesi - Diğer İlgi Çekici Metodlar

watchPosition() - Kullanıcının mevcut konumunu bildirir ve kullanıcı hareket ettiği sürece bildirmeye devam eder (arabalardaki GPS'ler gibi).

clearWatch() - Stops the watchPosition() method.

Aşağıdaki örnek watchPosition() metodunu gösterir. Bu örneği test edebilmemiz için hareketli halde olmanız gerekir (iPhone vs.):

Örnek

```
<script>
var x = document.getElementById("demo");
function getLocation()
{
  if (navigator.geolocation)
  {
    navigator.geolocation.watchPosition(showPosition);
  }
  else{x.innerHTML = "Tarayıcınız bu özelliği desteklemiyor.";}
}
function showPosition(position)
{
  x.innerHTML = "Enlem: " + position.coords.latitude +
  "<br>Boylam: " + position.coords.longitude;
}
</script>
```

[Kendin Dene »](#)

HTML5 Sürükle ve Bırak

◀ Önceki

Sonraki ▶

Sürükle ve Bırak, bir HTML5 standardıdır.



Yukarıdaki resmi dikdörtgenin içine sürükleyin.

Sürükle ve Bırak

Sürükle ve bırak çok bilinen ve ortak olan bir özelliktir. Bir nesneyi tutmak ve başka bir yere götürmek anlamına gelir.

HTML5'te herhangi bir element ya da nesne taşınabilir.

Tarayıcı Desteği



Internet Explorer 9+, Firefox, Opera, Chrome, ve Safari "Sürükle ve Bırak" özelliğini destekler.

Not: Safari 5.1.2. versiyonunda çalışmaz.

HTML5 Sürükle ve Bırak Örneği

Aşağıda basit bir örnek görülmektedir:

Örnek

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev)
{
ev.preventDefault();
}

function drag(ev)
{
ev.dataTransfer.setData("Text",ev.target.id);
}

function drop(ev)
{
ev.preventDefault();
var data = ev.dataTransfer.getData("Text");
ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>
```

```
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>


</html>
```

[Kendin Dene »](#)

Karışık görülebilir, o zaman sürükle ve bırak olayının diğer tüm farklı kısımlarına bir bakalım.

Bir elementi sürüklenebilir hale getirme

Her şeyden önce: Bir elementi sürüklenebilir hale getirmek için, "draggable" özelliğini "true" yapın:

```
<img draggable="true">
```

Neyi Sürükleyeceğiz? - ondragstart ve setData()

Sonra, element sürüklendiğinde ne olması gerektiğini belirtin.

Yukarıdaki örnekte, ondragstart özelliği drag(event) denilen, hangi data'nın sürükleneceğini belirten bir fonksiyonu çağırılmış.

DataTransfer.setData() metodu sürüklenen data'nın data tipini ve değerini ayarlar:

```
function drag(ev)
{
  ev.dataTransfer.setData("Text", ev.target.id);
}
```

Bu durumda, data tipi "text" ve değer (value) ise sürüklenebilen elementin id'sidir. ("drag1").

Nereye Sürükleyeceğiz? - ondragover

ondragover olayı (event) sürüklenen datanın nereye bırakılacağını (drop) belirtir.

Varsayılan olarak, veriler/elementler başka elementlerin içine bırakılamaz. Bir bırakma işlemine izin verebilmek için, elementin varsayılan kullanıma biçimini önlemeliyiz.

Bu iş ondragover event'i için event.preventDefault() metodu çağrılarak yapılır:

```
event.preventDefault()
```

Bırakma İşlemini Yapma - ondrop

Sürüklenen bir data bırakıldığında, bir "drop event" meydana gelir.

Yukarıdaki örnekte, ondrop özelliği drop(event) denilen bir fonksiyon çağırılmış:

```
function drop(ev)
{
  ev.preventDefault();
  var data = ev.dataTransfer.getData("Text");
  ev.target.appendChild(document.getElementById(data));
}
```

HTML5 Web Depolama

◀ Önceki

Sonraki ▶

HTML5 web depolama, çerezlerden (cookies) daha iyi.

HTML5 Web Depolama Nedir?

HTML5 ile, web sayfaları verileri kullanıcının tarayıcısına kaydedebilir.

Bu işlem önceden çerezler ile yapılırdı. Ancak Web Depolama (Web Storage) daha güvenli ve hızlıdır. Veri, her sunucu istemine eklenmez, sadece istendiği zaman kullanılır. Bu özellik ayrıca web sitesinin performansını etkilemeden çok büyük boyutlarda verinin saklanması sağlar.

Veri, isim/değer ikilileri şeklinde saklanır ve bir web sayfası sadece kendisi tarafından kaydedilen verilere ulaşabilir.

Çerezlerden farklı olarak, depolama limiti çok fazladır (en az 5 MB) ve bilgiler asla sunucuya gönderilmez.

Tarayıcı Desteği



Web Depolama Internet Explorer 8+, Firefox, Opera, Chrome ve Safari tarafından desteklenir.

HTML5 Web Depolama Nesneleri (Object)

HTML5 Web Depolama verileri istemci bilgisayarda saklamak için iki yeni nesne sağlar:

- window.localStorage - Verileri son kullanma tarihi olmadan saklama
- code.sessionStorage - Verileri bir oturum için saklama (Pencere kapatıldığında her şey silinir)

Web Depolama kullanmadan önce, tarayıcı tarafından desteklenip desteklenmediğini kontrol edin:

```
if(typeof(Storage)!="undefined")
{
  // localStorage/sessionStorage için kod kısmı.
}
else
{
  // Üzgünüm... Web Depolama desteği yok...
}
```

LocalStorage Nesnesi

LocalStorage nesnesi, son kullanma tarihi olmadan verileri depolar. Veriler, tarayıcı kapatıldığında silinmez ve sonraki gün, hafta hatta yıl bile kullanılabilir.

Örnek

```
// Kaydet
localStorage.setItem("lastname", "Smith");
// Geri al
document.getElementById("result").innerHTML=localStorage.getItem("lastname");
```

[Kendin Dene »](#)

Örneğim açıklaması:

- Önce name="lastname" ve value="Smith" ile bir localStorage isim/değer ikilisi oluşturun
- Sonra "lastname" içindeki değeri geri alın ve id="result" ile elementin içine ekleyin

Yukarıdaki örnek, şu şekilde de yazılabilir:

```
// Kaydet
localStorage.lastname = "Smith";
// Geri al
document.getElementById("result").innerHTML=localStorage.lastname;
```

"lastname" değerini localStorage'den kaldırmak için gereken sözdizimi (syntax) ise şöyledir:

```
localStorage.removeItem("lastname");
```

Not: İsim/değer (Name/value) ikilileri her zaman alfa-numerik (string) olarak kaydedilir. Gerekğinde, onları istediğiniz formata dönüştürmeyi unutmayın!

Aşağıdaki örnek, kullanıcının bir butona kaç kez tıkladığını sayıyor. Bu kodda, string olarak kaydedilen tıklanma sayısı, sayacın arttırabilmesi için Number() komutu ile numerik değere casting (dönüştürme) yapıyor:

Örnek

```
if (localStorage.clickcount)
{
    localStorage.clickcount = Number(localStorage.clickcount) + 1;
}
else
{
    localStorage.clickcount = 1;
}
document.getElementById("result").innerHTML="Siz " +
localStorage.clickcount + " kere butona tıkladınız.";
```

[Kendin Dene »](#)

SessionStorage Nesnesi

SessionStorage nesnesi sadece bir oturum için veri depolar. Bunun dışında tüm özellikleri localStorage ile aynıdır. Tarayıcı kapatıldığında veriler silinir.

Aşağıdaki örnek, mevcut oturum içinde bir butona kaç kez tıkladığını sayar:

Örnek

```
if (sessionStorage.clickcount)
{
    sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;
}
else
{
    sessionStorage.clickcount = 1;
}
document.getElementById("result").innerHTML="Bu oturumda " +
sessionStorage.clickcount + " kere butona tıkladınız.";
```

[Kendin Dene »](#)

HTML5 Uygulama Bellekleme

◀ Önceki

Sonraki ▶

HTML5 ile bir web uygulamasının çevrimdışı (offline) versiyonunu oluşturmak, bir cache manifest dosyası oluşturarak yapmak kolaydır.

Uygulama Bellekleme (Application Cache) Nedir?

HTML5 bir web uygulamasının ön belleğe alınmasını ve internet yokken de erişilebilmesini sağlayan "uygulama bellekleme" özelliğini taşır.

Application cache bir uygulamaya üç avantaj sağlar:

1. Çevrimdışı tarama - kullanıcılar çevrimdışı olduklarında da uygulamayı kullanabilir.
2. Hız - Belleklenmiş uygulamalar daha çabuk yüklenir.
3. Düşürülmüş sunucu yükü - tarayıcı sadece gerekli güncellemeleri/değişiklikleri sunucudan yükler.

Destekleyen Tarayıcılar



Internet Explorer 10, Firefox, Chrome, Safari ve Opera Uygulama Bellekleme'yi destekler.

HTML5 Cache Manifest Örneği

Aşağıdaki örnek cache manifest içeren bir HTML dökümanını gösterir (çevrimdışı tarama):

Example

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">

<body>
Döküman içeriği.....
</body>

</html>
```

[Kendin Dene »](#)

Cache Manifest İlk Bilgiler

Uygulama Bellekleme'yi aktifleştirmek için, manifest özelliğini <html> etiketinin içine ekleyin:

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">
...
</html>
```

Manifest özelliği olan her sayfa kullanıcı tarafından ziyaret edildiğinde ön belleğe alınır.

Manifest dosyası için önerilen dosya uzantısı: ".appcache"tir.



Bir manifest dosyası **correct MIME-type** ile sunucuya yüklenmelidir. Bu correct MIME-type "text/cache-manifest"tir. Web sunucusu üzerinden konfigüre edilmelidir.

Manifest Dosyası

Manifest dosyası, tarayıcıya neyi ön belleğe almasını (cache) gerektiğini söyleyen basit bir metin belgesidir.

Manifest dosyası üç kısımdan oluşur:

- **CACHE MANIFEST** - Bu başlığın altında listelenen dosyalar ilk kez download edildikten sonra ön belleğe alınır.
- **NETWORK** - Bu başlığın altında listelenen dosyalar sunucuya bağlantı gerektirir ve asla ön belleğe alınmazlar.
- **FALLBACK** - Bu başlığın altında listelenen dosyalar bir dosya erişilemez durumda ise kullanıcıları yönlendirmek istediğiniz bir sayfayı belirtir (örnek: 404 bulunamadı sayfası gibi)

CACHE MANIFEST

İlk satır CACHE MANIFEST olmalıdır ve zorunludur:

```
CACHE MANIFEST
/theme.css
/logo.gif
/main.js
```

Yukarıdaki manifest dosyası üç kaynak dosyasına sahiptir: bir CSS stil dosyası, bir GIF resim dosyası ve bir JavaScript dosyası. Bir manifest dosyası yüklendiğinde, tarayıcı bu üç dosyayı sunucudan download eder. Devamında, kullanıcı internete bağlı olmazsa bile bunlara erişebilir.

NETWORK

Aşağıdaki NETWORK kısmı "login.php" dosyasının asla ön belleğe alınmayacağını ve çevrimdışı olarak kullanılmayacağını belirtir.:

```
NETWORK:
login.php
```

Geri kalan tüm dosyaların internet bağlantısı gerektirdiğini belirtmek için bir yıldız (*) simgesi kullanılabilir:

```
NETWORK:
*
```

FALLBACK

Aşağıdaki FALLBACK kısmı "offline.html" dosyasının, herhangi bir internet kesintisi yaşandığında ya da erişilemez bir sayfa olduğunda kullanıcılara gösterilmek üzere /html/ klasörüne koyulacağını gösterir.:

```
FALLBACK:
/html/ /offline.html
```

Not: İlk [URI](#) kaynak, ikincisi ise fallback'tir.

Önbelleği Güncelleme (Update)

Bir uygulama bir kez ön belleğe alındıktan sonra aşağıdakilerden biri olana kadar aynı şekilde kalır:

- Kullanıcı tarayıcının ön belleğini temizler ise...
- Manifest dosyası değiştirilir ise... (örnek aşağıda)
- Uygulama ön belleği programlı bir şekilde güncellenir ise...

Örnek - Tüm bir Cache Manifest Dosyası

```
CACHE MANIFEST
# 2012-02-21 v1.0.0
/theme.css
/logo.gif
/main.js
```

NETWORK:
login.php

FALLBACK:
/html/ /offline.html



İpucu: "#" işareti ile başlayan satırlar yorum satırlarıdır, ancak başka bir amaca da hizmet ederler. Bir uygulamanın ön belleğe alınmış hali sadece manifest dosyası değiştirildiğinde güncellenir. Eğer bir resim ya da bir JavaScript fonksiyonunu düzenlerseniz (edit) bu değişiklikler tekrar ön belleğe alınmayacaktır. Yorum satırlarındaki tarih ya da versiyon gibi bilgileri değiştirmek ise dosyalarınızın tekrar ön belleğe alınması (update) için bir yoldur.

Uygulama bellekleme için notlar

Neyi ön belleğe aldığınıza dair dikkatli olun.

Bir dosya ön belleğe alındıktan sonra, siz dosyayı sunucu tarafında güncellerseniz bile tarayıcı belleğe alınmış şeklini göstermeye devam edecektir. Tarayıcınız güncellenmiş şeklini yüklediğinden emin olmak için, manifest dosyasını değiştirmek (yorum satırlarında ufak değişiklikler yapmak gibi) zorundasınız.

Not: Tarayıcıların farklı boyutlarda cache etme limitleri olabilir (bazıları site başına 5MB limit verir).

HTML5 Web Çalışanı (Workers)

◀ Önceki

Sonraki ▶

Web çalışanı, sayfanın performansını etkilemeden arka planda çalışan bir JavaScript kodudur.

Web Çalışanı Nedir?

HTML sayfasında bir script çalıştırıldığında, script bitene kadar sayfa kullanıcıya cevap veremez hale gelir.

Web çalışanı, sayfanın performansını etkilemeden diğer script'lerden bağımsız olarak arka planda çalışan bir JavaScript kodudur. İsteddiğiniz başka her türlü şeyi yapmaya devam edebilirsiniz: tıklama, nesnelere seçme vs...

Tarayıcı Desteği



Internet Explorer 10, Firefox, Chrome, Safari ve Opera destekler.

HTML5 Web Workers Örneği

Aşağıdaki örnek, arka planda sayı sayan basit bir web çalışanı oluşturur:

Örnek

Saymaya Başla:

Çalıştır

Durdur

Kendin Dene »

Tarayıcı Desteğini Kontrol Etme

Bir web worker oluşturmadan önce bu işlemi mutlaka yapın:

```
if(typeof(Worker) !== "undefined")
{
  // Evet! Web Worker Destekleniyor!
  // Kod satırları.....
}
else
{
  // Tarayıcınız Web Worker özelliğini desteklemiyor.
}
```

Bir Web Worker Dosyası Oluşturma

Şimdi, kendi harici JavaScript web worker dosyamızı oluşturalım.

Burda, sayı sayan bir script oluşturuyoruz. Script, "demo_workers.js" dosyası içinde saklanacak:

```
var i = 0;
```

```
function timedCount()
{
i = i + 1;
postMessage(i);
setTimeout("timedCount()",500);
}

timedCount();
```

Yukarıdaki kodda önemli olan kısım, HTML dosyasına mesaj göndermek için kullanılan **postMessage()** metodudur.

Not: Normalde web çalışanları böyle basit scriptler için kullanılmaz. Daha çok yoğun ve ağır CPU (işlemci) işlemleri için kullanılır.

Bir Web Worker Nesnesi Oluşturmak

Artık bir web worker dosyamız var, şimdi yapmamız gereken onu bir HTML dosyası ile çağırmak.

Aşağıdaki satırlar bir worker olup olmadığını kontrol eder, eğer yoksa yeni bir tane yaratır ve "demo_workers.js" içindeki kodları çalıştırır:

```
if(typeof(w) == "undefined")
{
w = new Worker("demo_workers.js");
}
```

Artık worker'dan mesaj alabilir ya da worker'a mesaj gönderebiliriz.

Web Çalışanına bir tane "onmessage" olay dinleyicisi (event listener) ekleyin.

```
w.onmessage = function(event){
document.getElementById("result").innerHTML = event.data;
};
```

Web çalışanı bir mesaj gönderdiğinde, event listener içindeki kod çalışır. Web çalışanından alınan veri, event.data içerisinde saklanır.

Web Çalışanını Bitirmek / Silmek

Bir Web Worker nesnesi oluşturulduğunda, silinene kadar mesajları dinlemeye (harici script sonradan kapatılsa bile) devam eder.

Bir web çalışanını silmek ve tarayıcının/bilgisayarın kaynaklarını serbest bırakmak için terminate() metodunu kullanın:

```
w.terminate();
```

Web Çalışanı ve DOM (Document Object Model - Döküman Nesne Modeli)

Web Çalışanları harici dosyalar oldukları için, aşağıdaki JavaScript nesnelere ulaşamazlar:

- window object
- document object
- parent object

HTML5 Server-Sent İşlemleri

◀ Önceki

Sonraki ▶

HTML5 SSE bir web sayfasının sunucudan güncellemeler almasını sağlar.

Server-Sent İşlemleri - Tek Yönlü Mesaj Gönderme

Bir Server Sent işlemi, bir web sayfası otomatik olarak sunucudan güncelleme alınca gerçekleşir.

Bu önceden de mümkündü ancak web sayfası herhangi bir güncelleme olup olmadığını sormak zorundaydı. SSE ile güncellemeler otomatik olarak yüklenir.

Örnek: Facebook/Twitter güncellemeleri, yeni haber uyarıları, maç sonuçları vs.

Tarayıcı Desteği



SSE Bildirimlerini Almak

EventSource nesnesi sunucu bildirimlerini almak için kullanılmıştır:

Örnek

```
var source = new EventSource("demo_sse.php");
source.onmessage = function(event) {
    document.getElementById("result").innerHTML += event.data + "<br>";
};
```

[Kendin Dene >>](#)

Örneğin açıklaması:

- Yeni bir EventSource nesnesi oluştur ve güncellemeler için gerekli sayfanın URL'sini belirt.(Bu örnekte "demo_sse.php")
- Her güncelleme alındığında onmessage event'i (olayı) meydana gelir
- Bir onmessage olayı meydana geldiğinde, alınan veriyi id="result" ile elementin içine koy

Server-Side (Sunucu Tarafı) Kodu Örneği

Yukarıdaki örneğin çalışması için, veri güncellemeleri gönderebilen bir sunucuya ihtiyacınız vardır (PHP ya da ASP gibi)

PHP Kodu (demo_sse.php):

```
<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');

$time = date('r');
echo "data: Sunucu saati: {$time}\n\n";
flush();
?>
```

ASP Kodu(demo_sse.asp):

```
<%
Response.ContentType = "text/event-stream"
```

```
Response.Expires = -1
Response.Write("data: " & now())
Response.Flush()
%>
```

Kodun açıklaması:

- "Content-Type"ı başlık olarak "text/event-stream"e ayarla
- Sayfanın ön bellek yapmayacağını belirt.
- Sonucu hazırla (**Her zaman** "data: " ile başla)
- Çıkış bilgisini web sayfasına gönder

EventSource Nesnesi

Yukarıdaki örneklerde mesajları alabilmek için onmessage event'ini kullandık. Ama başka event'lar da vardır:

Olay (Events)	Açıklama
onopen	Sunucuya bir bağlantı açıldığında
onmessage	Bir mesaj alındığında
onerror	Bir hata oluştuğunda