



JQUERY'E GİRİŞ



Atatürk Üniversitesi
Açıköğretim Fakültesi

İNTERNET PROGRAMCILIĞI I Öğr. Gör. Rafet Orçun MADRAN

İÇİNDEKİLER



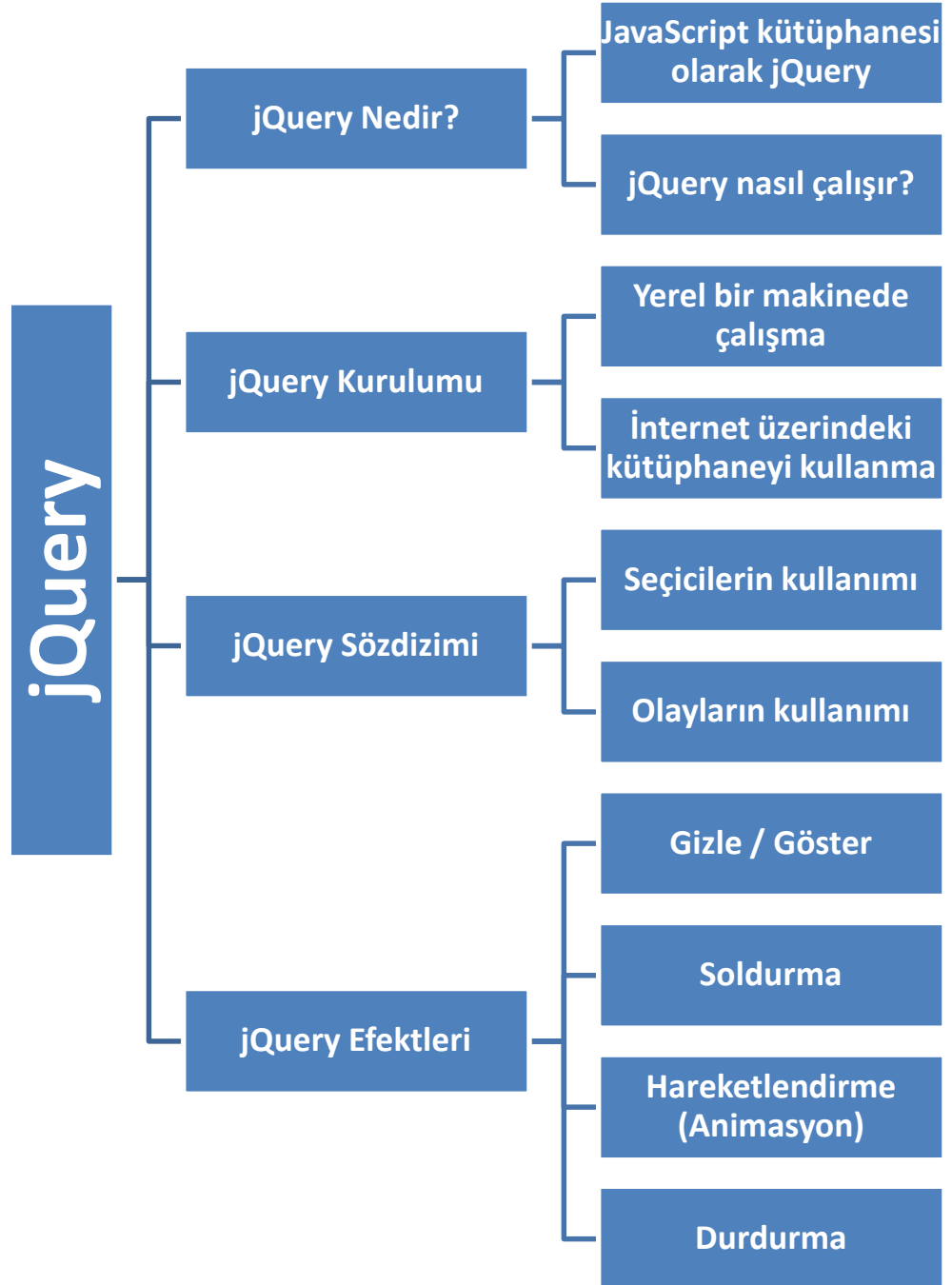
- jQuery Nedir?
 - Bir JavaScript Kütüphanesi olarak jQuery
 - Nasıl Çalışır?
- jQuery Kurulumu
- jQuery Sözdizimi (Syntax)
 - Seçiciler (Selectors)
 - Olaylar (Events)
- jQuery Efektleri

HEDEFLER



- Bu üniteyi çalıştıktan sonra;
 - jQuery kütüphanesinin kurulumunu yapabilecek,
 - jQuery'nin genel sözdizimi ve kullanımı ile ilgili bilgi sahibi olabilecek,
 - jQuery'deki seçicileri kullanabilecek,
 - jQuery'de olaylar oluşturabilecek,
 - HTML elemanları ile animasyonlar yapabileceksiniz.

ÜNİTE 12



GİRİŞ

Web teknolojileri HTML standartlarının ilk kullanılmaya başlandığı 1993 yılından bu yana inanılmaz bir hızla gelişti. Özellikle farklı hizmet sektörlerinde yer alan birçok uygulamanın İnternet tabanlı olarak kullanılmaya başlaması hem tarayıcıların yeteneklerinin gelişmesine hem de kullanılan teknolojilerde çeşitliliğe yol açtı. Aslında bu gelişmeyi tetikleyen, tarayıcı üzerinde çalışan İnternet tabanlı uygulamalardan masaüstü bilgisayarlarda çalışan uygulamaların performansının beklenmesiydi. Son kullanıcılar, bilgisayarlarına ek bir yazılım kurmadan tüm işlemleri İnternet tarayıcıları üzerinden gerçekleştirmek istiyorlardı.

HTML tek başına yukarıda belirttiğimiz uygulama deneyimini son kullanıcıya yaşatabilecek bir altyapıya sahip değildir. Bu noktada devreye giren istemci-sunucu mimarisinin ürettiği çözümler de yeterli gelmemektedir; her bir işlem adımı Web sayfasının yenilenmesine yol açan geleneksel HTML altyapısı, kullanıcıya sunulmak istenen hızı ve esnekliği sağlamamaktadır. Çözüm, Web sayfalarının yenilenmesine gerek kalmadan da veri alışverişini ve etkileşimi mümkün kılacak teknolojilerde yatmaktadır. Ajax teknolojileri olarak ifade edilen bu mimarinin en yaygın kullanılan kütüphanelerinden biri JQuery'dir.

Bu ünite, Web programlama içerisinde ihtiyaç duyulabilecek navigasyon, animasyon ve sayfa içi etkileşimleri oluşturabilmenizi sağlayacak kodları çok kolay bir şekilde yazabilmenizi sağlayacak araçları size sunacaktır. Bu çalışmaları gerçekleştirebilmek için kullanacağımız JQuery kütüphanesi hakkında temel bilgiler, kurulum, kullanım şekilleri ve örnek uygulamalar da yine bu ünite içinde yer alacaktır.

jQuery NEDİR?

jQuery, hızlı, küçük ve zengin özelliklere sahip bir JavaScript kütüphanesidir. Çok sayıda tarayıcıda çalışan kullanımı kolay bir API (Application Programming Interface / Uygulama Programlama Arayüzü) ile HTML belgesinde gezinme, olay işleme, animasyon ve diğer Ajax uygulamalarını çok daha basit hâle getirir. Çok yönlü ve genişletilebilir yapısı çok fazla sayıda geliştirici tarafından tercih edilmesine yol açmaktadır.

jQuery, açık kaynak kodlu bir projedir. Proje ile ilgili tüm bilgilere <http://jquery.com/> adresinden ulaşılabilir. Projenin tüm kodları MIT lisansı ile kullanıma sunulmuştur. MIT lisansının içeriği ve kullanım ile ilgili detaylı bilgi <http://ozgurlisanslar.org.tr/mit/> adresinde yer almaktadır.

Bir JavaScript Kütüphanesi Olarak jQuery

Programlama dillerinin standart yapılarına ek olarak kullanılabilecek özellikleri içinde barındıran kod bloklarını *kütüphane* olarak tanımlayabiliriz. Kütüphaneler belirli bir amaç çerçevesinde birçok tanımlamayı ve hazır fonksiyonu içlerinde bulunduran yapılar olarak kod yazma süreçlerini çok kolaylaştırmakta ve geliştiricilere hızlı uygulama geliştirme olanağı sunmaktadırlar.



jQuery, hızlı, küçük ve zengin özelliklere sahip açık kaynak kodlu bir JavaScript kütüphanesidir.

Slogan olarak “*write less, do more / az yaz, çok iş yap*” cümlesini kullanan jQuery, özellikle sayfa içi etkileşim ve basit animasyonların gerçekleştirilebilmesini çok kolay hâle getiren bir JavaScript kütüphanesidir.

jQuery Nasıl Çalışır?



jQuery kütüphanesi, yerel bir adreste yer alabileceği gibi, İnternet üzerindeki bir depodan da kullanılabilir.

JavaScript kütüphanelerinin genel çalışma prensipleri jQuery için de geçerlidir. Aslında basit bir metin dosyasından ibaret olan kütüphane dosyasına HTML kodunun yazıldığı dosyadan bağlantı verilmesi yeterli olmaktadır. jQuery kütüphane dosyasına iki farklı şekilde bağlantı sağlanabilir:

- Kütüphane dosyası yerel bir adreste bulunabilir (çalıştığınız bilgisayarın sabit diskinde ya da sunucunuzun herhangi bir dizininde).
- Kütüphane dosyası jQuery'nin İnternet üzerinde yer alan depolarından kullanılabilir.

Yukarıda belirttiğimiz bağlantı şekilleri aynı zamanda jQuery'nin nasıl kurulacağı ile ilgili durumu da belirler. jQuery kütüphane dosyasının *sürekli güncel kalması* isteniyorsa kütüphanenin İnternet üzerindeki depodan kullanılması uygun olacaktır. Ancak eğer geliştirilecek uygulama İnternet'e bağlı olmayan bir platform için ise (örneğin bir kiosk sistemi ya da bir gömülü sistem) kütüphane dosyasının yerel bir adreste bulunması gerekir.

jQuery'nin nasıl çalıştığını gösteren kod bloku Şekil 12.1.'de gösterilmektedir. Şekil 12.1.'in 9. satırında yer alan adres tanımlaması kütüphane dosyasının yerel ya da İnternet üzerindeki bulunmasına bağlı olarak değişiklik gösterebilir.

```

1. <!doctype html>
2. <html>
3. <head>
4.     <meta charset="utf-8">
5.     <title>Örnek Bağlantı</title>
6. </head>
7. <body>
8.     <a href="http://jquery.com/">jQuery</a>
9.     <script src="jquery.js"></script>
10.    <script>
11.
12.        // jQuery kodları burada yer alacak!
13.
14.    </script>
15. </body>
16. </html>

```

Şekil 12.1. JQuery'nin Çalışma Şekli, Örnek Kod Bloku.

jQuery KURULUMU

jQuery ile ilgili kurulumda ihtiyaç duyacağımız tüm kaynak dosyalar <http://jquery.com/download/> adresinde yer almaktadır. Bu sayfada jQuery dosyalarının hem sıkıştırılmış hem de sıkıştırılmamış sürümleri yer alır.

Sıkıştırılmamış dosya biçimi daha çok **kod üzerinde geliştirme çalışması** yapmak isteyenler için hazırlanmıştır; okunması, üzerinde değişiklik yapılması daha kolaydır. Sıkıştırılmış dosya biçiminin avantajı ise **boyut olarak daha küçük** olmasıdır. Bu şekilde HTML sayfalarının İnternet tarayıcıları tarafından yüklenmesi daha hızlı olur.



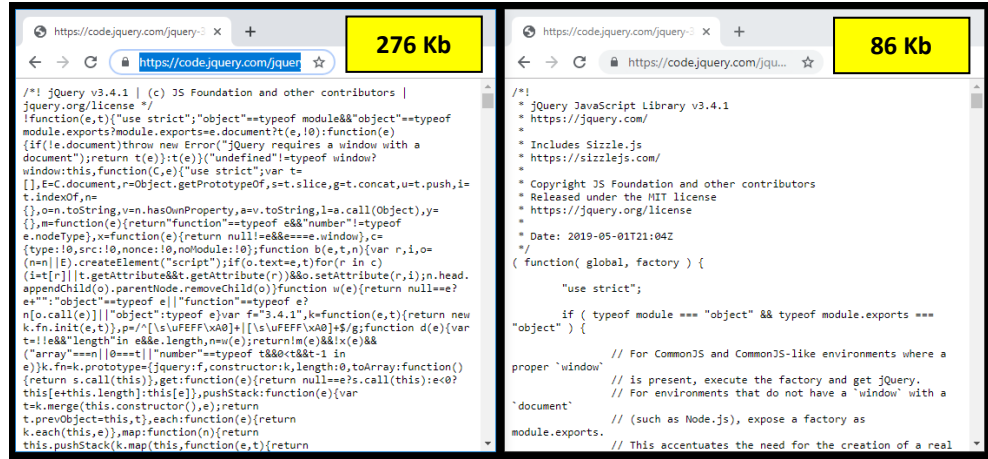
jQuery çekirdek dosyası
iki farklı biçimde
kullanılabilir:
sıkıştırılmış ya da
sıkıştırılmamış.

jQuery kullanımınız kütüphanenin hazır olan özelliklerini kullanmaktan ibaret olacaksa ve çekirdek kod üzerinde herhangi bir değişiklik yapmayacaksanız, her zaman sıkıştırılmış jQuery sürümlerini tercih etmeniz gerekir. Bu şekilde çalışmanız Web sitesinin genel optimizasyonu açısından önem taşımaktadır.

jQuery dosya isimlendirmelerinde belirli bir algoritma kullanılmaktadır. Bu sayede dosyanın adından ana sürüm bilgisi, alt sürüm bilgileri ve dosyanın biçimi hakkında bilgiler elde edilebilir. Aşağıdaki listede jQuery'nin son kararlı sürümü olan 3.4.1 için dosya isimlendirmeleri yer almaktadır:

- jQuery 3.4.1 geliştirme sürümü, sıkıştırılmış: jquery-3.4.1.min.js
- jQuery 3.4.1 üretim sürümü, sıkıştırılmamış: jquery-3.4.1.js

Şekil 12.2.'de jQuery'nin çekirdek kodlarının sıkıştırılmış (sol) ve sıkıştırılmamış (sağ) dosya sürümleri karşılaştırma yapabilmemiz için yan yana yer almaktadır. Şekil 12.2.'de de görüldüğü gibi sıkıştırılmamış olan dosya biçimi **çok daha rahat** okunabilmektedir, ancak dosya boyutu açısından önemli bir fark bulunmaktadır. Özellikle yavaş internet bağlantılarında (mobil cihazlar vb. gibi) sayfanın yüklenme hızını etkileyebilir.



Şekil 12.2. Jquery Sıkıştırılmış (Sol) Ve Sıkıştırılmamış (Sağ) Dosya Biçimleri Ve Boyutları.

Şimdi jQuery'nin sıkıştırılmış dosya biçimindeki kütüphane dosyasını (jquery-3.4.1.js) sabit diskimizde oluşturduğumuz bir proje klasörü içerisine <http://jquery.com/download/> adresinden yükleyerek kaydedelim. Kaydetme işlemi sırasında kütüphane dosyasına istediğiniz bir ismi verebilir ya da mevcut ismi değiştirmeden kullanabilirsiniz. Bu çalışmada dosya ismimizi "jquery.js" olarak sadeleştirebiliriz.

Şekil 12.3.'te <http://jquery.com/download/> adresinde jQuery başlığı altında yer alan "Download the compressed, production jQuery 3.4.1" bağlantısı

görülmektedir. Bu bağlantıya sağ tıklayarak “**bağlantıyı farklı kaydet**” seçeneği ile dosyayı yerel diskinizde oluşturduğunuz “jQuery” proje klasörünün içine kaydediniz.

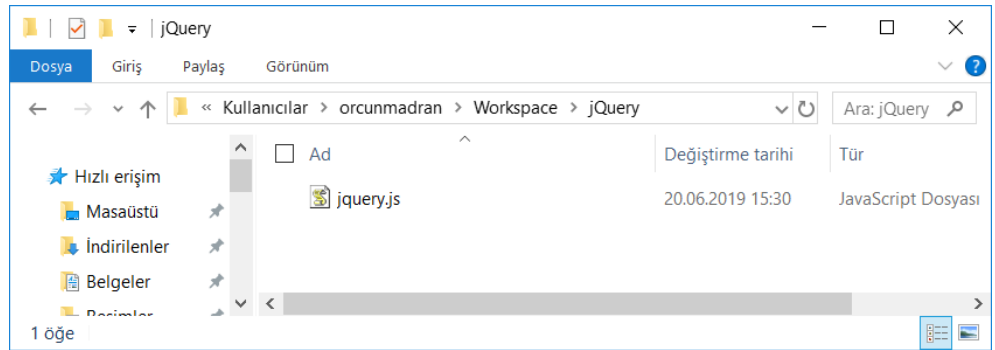


İnternet üzerinde yer alan JavaScript dosyalarını, dosya bağlantısının üzerine sağ klik ile tıklayarak “dosyayı farklı kaydet” yardımı ile diskinize kaydedebilirsiniz.



Şekil 12.3. JQuery Yükleme Sayfası Ve Kütüphane Dosyası Bağlantısı.

Kayıt işlemini gerçekleştirdikten sonra jQuery adlı proje klasörümüzün görünümü Şekil 12.4.’teki gibi olmalıdır.



Şekil 12.4. Sabit Diskimizde Yer Alan “Jquery” Adlı Proje Klasörü.

Kurulum işleminin dosya yapılandırma aşamasını tamamladığımıza göre artık test için bir HTML dokümanı oluşturup jQuery kütüphane dosyası ile bağlantısını kurabiliriz. Oluşturacağımız HTML dokümanının ismi “test.html” ve içeriği de Şekil 12.5.’teki gibi olsun.

```

1. <!doctype html>
2. <html>
3. <head>
4.     <meta charset="utf-8">
5.     <title>jQuery Test</title>
6.     <script src="jquery.js"></script>
7.     <script>
8.         $(document).ready(function() {
9.             $("p").click(function() {
10.                 $(this).hide();
11.             });
12.         });
13.     </script>

```

```

14. </head>
15. <body>
16.     <p>Tıkla ve gizle..!</p>
17. </body>
18. </html>

```

Şekil 12.5. JQuery Test Dokümanı İçinde Yer Alan Kod Bloku.

Şekil 12.5.'teki kodu incelediğimizde başlık (<head>) bölümü içerisinde 6. satırda jQuery kütüphanesi ile bağlantı kurulduğunu görüyoruz. 7. ve 13. satırlar arasında ise JavaScript kodlarımız yer almakta. Bu test çalışmasında sayfanın gövde bölümü (<body>) içerisinde yer alan <p> etiketi ile girilmiş metinlerin üzerine tıklandığı zaman metnin gizlenmesini (bir anlamda sayfadan kaybolmasını) amaçlıyoruz.

Eğer yukarıdaki adımları doğru bir şekilde yaptıysanız, test.html dosyasını herhangi bir İnternet tarayıcısında açtığınızda "Tıkla ve gizle..!" metnine tıklayıp yok olduğunu görebilmeniz gerekir.



İçerik Dağıtım Ağı (CDN) sistemi ile jQuery kütüphanelerini sabit diskinize kaydetmeden, İnternet üzerinden de kullanabilirsiniz.

Bu örneğimizi jQuery kütüphane dosyasına İnternet üzerindeki bir kaynaktan bağlantı sağlayarak da test edebiliriz. Bu sayede ilgili kütüphane sürümünün *her zaman en güncel* hâlini kullanmış oluruz. Şekil 12.5.'te yer alan test kodumuzun 6. satırını aşağıdaki gibi değiştirelim. Bu değişikliği yaptıktan sonra test.html dosyasını tarayıcıda açalım. İnternet bağlantımızın olması gerektiğini de unutmayalım.

```

6. <script src="http://code.jquery.com/jquery-3.4.1.min.js"></script>

```

CDN (Content Delivery Network) yani İçerik Dağıtım Ağı adı verilen bu sistem ile birçok farklı kaynak kod sabit diske kaydedilmeden de kullanılabilir. jQuery'nin CDN sistemine <http://code.jquery.com/> adresinden ulaşılabilir ve farklı sürümleri bu sayfa aracılığıyla kullanabilirsiniz.

jQuery ile ilgili kurulum aşamaları tamamlandı. Artık jQuery ile ilgili daha detaylı çalışmaları yapabilmek için gerekli altyapıya sahipsiniz.



Bireysel Etkinlik

- Sabit diskinizde "jQuery" adlı bir proje klasörü oluşturun.
- Proje klasörünüzün içine jQuery kütüphane dosyasını "jquery.js" olarak Şekil 12.3.'teki bağlantıyı kullanarak Şekil 12.4.'teki gibi kaydedin.
- Şekil 12.5.'te yer alan kodları test.html dosyasına yerleştirerek İnternet tarayıcınızda görüntüleyin.

jQuery SÖZDİZİMİ (Syntax)

jQuery sözdizimi HTML öğelerinin (etiketlerinin) ve CSS tanımlamalarının

seçilmesi ve bu öğelere birtakım eylemler atanması şeklinde gerçekleştirilir.



jQuery, sayfa henüz tam olarak yüklenmeden kodların çalışmaya başlamasını engellemek için *doküman hazır olayı* adında bir fonksiyon kullanır.



Örnek

- Sözdizimi şablonu: **\$(seçici).eylem()**
- Mevcut ögenin gizlenmesi: **\$(this).hide()**
- <p> etiketli tüm öğelerin gizlenmesi: **\$("#p").hide()**
- "test" sınıfına dâhil tüm öğelerin gizlenmesi: **\$(".test").hide()**
- "test" kimliğine sahip tüm öğelerin gizlenmesi: **\$("#test").hide()**

jQuery sözdizimi içerisinde önemli noktalardan biri de “The Document Ready Event” adı verilen *doküman hazır olayı*dır. Bu olay genel kullanımda jQuery kodlarını kapsayıcı bir fonksiyondur. Kullanılan bu fonksiyon sayesinde dokümanın yüklenmesi henüz tamamlanmadan jQuery kodlarının çalışması engellenir ve olası bir hatadan kaçınılmış olur.

Doküman hazır olayının kullanımı Şekil 12.6.’da gösterilmiştir. Rahat okunması amacıyla çoklu satır olarak görüntülenen bu kod bloku aslında tek satır olarak yorumlanır.

```
1. $(document).ready(function() {  
    // jQuery kodları bu bölümde yer alacak.  
});
```

Şekil 12.6. Doküman Hazır Olayı Örnek Kodları.

jQuery geliştirici takımı bu olay için daha kısa bir metot da oluşturmuştur. Kısa metot Şekil 12.7.’de gösterilmektedir.

```
$(function(){  
    // jQuery kodları bu bölümde yer alacak.  
});
```

Şekil 12.7. Doküman Hazır Olayı Kısaltılmış Sürümü.

Şekil 12.6. ve Şekil 12.7.’deki kod bloklarından hangisini tercih edeceğiniz size kalmıştır. Genel olarak tavsiye edilen ve bu ünite içerisindeki kullanılan metot Şekil 12.6.’daki metot olacaktır.

jQuery Seçicileri (Selectors)

jQuery seçicileri HTML öğelerinin seçimini ve bu öğeler ile ilgili değişiklikleri mümkün kılar. Seçiciler HTML öğelerini adlarına, kimliklerine (id), sınıflarına (class), tiplerine, niteliklerine, niteliklerinin sahip olduğu değerlere göre seçebilirler.

Daha önce test dokümanımızda örneğini gördüğümüz gizleme olayını ele alalım. Bu olay sayfadaki herhangi bir HTML ögesini gizlememize olanak sağlar. HTML ögesi olarak paragraf tanımlayıcısı olan "p" etiketini seçelim. Örnek kod Şekil 12.8.'deki gibi olmalıdır.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.     $("button").click(function() {
8.         $("p").hide();
9.     });
10. });
11. </script>
12. </head>
13. <body>
14.
15. <h2>jQuery'e Hoşgeldiniz</h2>
16.
17. <p>jQuery bir JavaScript (JS) kütüphanesidir.</p>
18. <p>Hızlı ve kolay JS kodlama yapmanızı sağlar!</p>
19.
20. <button>Paragrafları gizlemek için tıklayınız</button>
21.
22. </body>
23. </html>

```



jQuery seçicileri, aynen CSS sözdiziminde olduğu gibi HTML ögelerini etiketleri, sınıfları ya da kimlikleri ile seçebilirler.

Şekil 12.8. JQuery Seçici Örnek Kodu.

Şekil 12.8.'deki kodu incelediğimizde 6. satırda öncelikli olarak tüm sayfanın yüklenmesini bekleyen fonksiyonu görüyoruz. Bu fonksiyondan sonra 8. satırda HTML ögesi olarak "p" seçilmiş durumda; sayfa içerisinde <p> etiketi ile tanımlanmış olan tüm metinleri etkileyecek bir seçim.

Eğer 8. satırda yer alan seçiciyi "p" yerine "h2" olarak değiştirirsek, bu sefer gizlenecek olan öge ikinci seviyeden başlık olan <h2> etiketi ile tanımlanmış metin



Bireysel Etkinlik

- Proje klasörünüzün içinde "secici.html" adlı bir dosya oluşturun.
- Şekil 12.8.'de yer alan kodları secici.html dosyasına yerleştirerek internet tarayıcıda görüntüleyin.
- "secici.html" içinde yer alan kodlarda jQuery seçicisini <h2> etiketini seçecek şekilde yeniden düzenleyin ve test edin.

olacaktır.

jQuery'nin birçok farklı şekilde seçim işlemini gerçekleştirebileceğini belirtmiştik. Bir HTML sayfası içinde yer alan tüm paragrafları değil, sadece belirlediğimiz paragrafları gizlemek istiyoruz. Bu noktada bir HTML etiketi değil, etiketin içindeki özel bir sınıfı seçim işlemi için kullanabiliriz. Örnek uygulama Şekil

12.9.'daki gibi kodlanmalıdır.



Bir HTML sayfası içerisinde belirli bir öge grubunun içinde sınıfları kullanarak özel seçimler yapabilirsiniz.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6.     $(document).ready(function() {
7.         $(".button").click(function() {
8.             $(".p.ilk").hide();
9.         });
10.    });
11. </script>
12. </head>
13. <body>
14.     <h2>jQuery'e Hoşgeldiniz</h2>
15.     <p class="ilk">jQuery bir JavaScript (JS)
        kütüphanesidir.</p>
16.     <p>Hızlı ve kolay JS kodlama yapmanızı sağlar!</p>
17.     <button>İlk paragrafı gizlemek için
        tıklayınız</button>
18. </body>
19. </html>
```

Şekil 12.9. Jquery Sınıf Seçimi Örnek Kodları.

jQuery Olayları

jQuery olayları, HTML sayfası içindeki etkileşimlere karşılık vermemizi mümkün kılar. Bu etkileşimler farenin bir HTML ögesi üzerinden geçmesi, bir form elemanın seçili hâle getirilmesi ya da HTML sayfası içerisindeki herhangi bir ögeye tıklanması şeklinde ortaya çıkabilir. Tablo 12.1.'de Web sayfalarında sıkça kullanılan olayların jQuery içerisinde kullanılan kodları yer almaktadır.

Tablo 12.1. Web Sayfalarında Sıkça Kullanılan Olaylar.

Fare Olayları	Klavye Olayları	Form Olayları	Doküman Olayları
click (tıklama)	keypress (tuşa basma)	submit (gönderme)	load (yükleme)
dblclick (çift tıklama)	keydown (tuş basılı)	change (değiştirme)	size (boyutlandırma)
mouseenter (fare girişi)	keyup (tuşu bırakma)	focus (odaklanma)	scroll (kaydırma)
mouseleave (fare çıkışı)		blur (bulanıklaşma)	unload (çıkma)

Bir sayfadaki paragraflara tıklandığı zaman herhangi bir fonksiyonun tetiklenmesi isteniyorsa kullanılacak kod yapısı Şekil 12.10.'daki gibi olmalıdır.

```
$(".p").click(function() {

    // tetiklenecek kodlar burada yer alacak

});
```

Şekil 12.10. Jquery Olay Yapısı.

jQuery'deki olay yapısını daha iyi anlayabilmek için bir örnek üzerinde çalışalım. Örnek uygulamamızda Web sayfası içerisinde yer alan paragrafların üzerine fare ile geldiğimizde bize her paragraf için farklı bir uyarı versin. Bu

çalışmanın kodları Şekil 12.11.'deki gibi olmalıdır.



Bir olay birden fazla
fonksiyonu
tetikleyebilir.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("#p1").mouseenter(function() {
8.     alert("1. paragrafa giriş yaptınız!");
9.   });
10.  $("#p2").mouseenter(function() {
11.    alert("2. paragrafa giriş yaptınız!");
12.  });
13. });
14. </script>
15. </head>
16. <body>
17. <p id="p1">Fare ile bu paragrafın üzerine gelin.</p>
18. <p id="p2">Fare ile 2. paragrafın üzerine gelin.</p>
19. </body>
20. </html>

```

Şekil 12.11. JQuery Olay Yapısı Örnek Uygulama.

Şekil 12.11.'deki kodları incelediğinizde daha önce Tablo 12.1.'de fare olayı olarak listelediğimiz “mouseenter” (fare girişi) olayının kullanıldığını göreceksiniz. Bu olay ilgili paragrafın üzerine fare ile gelindiğinde bir uyarı kutusunun görüntülenmesini tetiklemektedir. 7. ve 10. satırlardaki “mouseenter” olayı farklı bir olay ile değiştirildiğinde (örneğin “click”) olayın tetiklenme şekli de tahmin edebileceğiniz gibi değişecektir.



**Bireysel
Etkinlik**

- Proje klasörünüzün içinde "olay.html" adlı bir dosya oluşturun.
- Şekil 12.11.'de yer alan kod yapısını örnek alarak sayfada yer alan bir resim ögesinin üzerine çift tıklandığında bir uyarı mesajı çıkmasını sağlayacak kodları olay.html dosyası içinde yazın ve test edin.

jQuery EFEKTLERİ

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

- Gizle / Göster (Show / Hide)
- Soldurma (Fade)



jQuery efektlerini
kullanarak etkileyici
Web sayfaları
oluşturabilirsiniz.

- Kaydırma (Slide)
- Hareketlendirme (Animate)
- Durdurma (Stop)

Gizle / Göster (Show / Hide)

jQuery ile HTML elemanlarını `hide()` ve `show()` metotları ile gizleyebilir ya da görünür hâle getirebilirsiniz. `hide()` ve `show()` metotlarının sözdizimi Şekil 12.12.'de görüldüğü gibidir.

```
1. $(seçici).hide(gizleme hızı);
2. $(seçici).show(gösterme hızı);
```

Şekil 12.12. Hide() Ve Show() Metotlarının Sözdizimi

`hide()` ve `show()` metotlarının kullanımını için bir örnek bir uygulama yapalım. Örnek uygulamamızda sayfa içerisindeki bir paragrafı “gizle” butonu ile gizleyip, “göster” butonu ile görünür hâle getirelim. Bu çalışmanın kodları şekil 12.13.'teki gibi olmalıdır.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function(){
7.   $("#hide").click(function(){
8.     $("p").hide(1000);
9.   });
10.  $("#show").click(function(){
11.    $("p").show(1000);
12.  });
13. });
14. </script>
15. </head>
16. <body>
17. <p>jQuery ile paragraf gizle, göster!</p>
18. <button id="hide">Gizle</button>
19. <button id="show">Göster</button>
20. </body>
21. </html>
```

Şekil 12.13. Hide() Ve Show() Örnek Uygulama.

Örnek uygulamamızın 8. ve 11. satırlarında yer alan “hide” ve “show” fonksiyonlarının içinde rakam ile 1000 tanımlaması yapıldığı gözünüzden kaçmamıştır. Bu tanımlama gizleme ve gösterme işleminin 1 saniyede (1000 milisaniye) gerçekleşmesi komutunu fonksiyona göndermektedir. Fonksiyona herhangi bir değer gönderilmemesi durumunda işlem anında gerçekleşmektedir.



jQuery efektlerinin gerçekleşme hızı fonksiyonlar içinde milisaniye olarak tanımlanabilmektedir.



Bireysel Etkinlik

- Proje klasörünüzün içinde "gizleGoster.html" adlı bir dosya oluşturun.
- Şekil 12.13.'te yer alan kod yapısını örnek olarak sayfada yer alan bir paragrafı butonlar yardımıyla gizlemeye ve göstermeye çalışın.

jQuery efektleri için kullanılan metotlarda *"Toggle"* adlı özel bir metot türü bulunmaktadır. Bu metot türü ile tek bir tetikleyici kullanarak (örneğin Web sayfasında yer alan tek bir buton yardımıyla) işlemin ilk tetiklemede (fare örneğinde tıklamada) gerçekleşip, ikinci tetiklemede ise (fare örneğinde yeniden tıklamada) işlemin geri alınması sağlanabilmektedir. Şekil 12.14.'te Web sayfasında yer alan butona ilk tıklanıpta paragraf gizlenecek, aynı butona tekrar basıldığında ise paragraf görünür hâle gelecektir.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("button").click(function() {
8.     $("p").toggle();
9.   });
10. });
11. </script>
12. </head>
13. <body>
14. <button>Hem gizle, hem göster!</button>
15. <p>Bu paragraf butona ilk tıklandığında gizlenecek,
    butona ikinci tıklandığında görünür hale gelecek</p>
16. </body>
17. </html>

```

Şekil 12.14. JQuery Metotlarında "Toggle" Kullanımı.



Bireysel Etkinlik

- Proje klasörünüzün içinde "toogle.html" adlı bir dosya oluşturun.
- Şekil 12.14.'te yer alan kod yapısını örnek olarak sayfada yer alan bir paragrafı tek bir buton yardımıyla gizlemeye ve göstermeye çalışın.

Soldurma (Fade)

jQuery ile HTML elemanlarını "fade" metodunu kullanarak soldurma efekti ile gizleyebilir (fadeOut) ya da belirginleştirme efekti ile görünür hâle (fadeIn) getirebilirsiniz. jQuery içerisinde "fade" metodunun farklı kullanım şekilleri yer

almaktadır. Bunlar;

- **fadeIn()**: Belirginleştirerek gösterme,
- **fadeOut()**: Soldurarak gizleme,
- **fadeToggle()**: Soldurarak gizleme ve belirginleştirerek gösterme,
- **fadeTo()**: Ögeyi soldururken belirli bir saydamlık değeri verme.

“fade” metotlarının sözdizimleri Şekil 12.15.’te görüldüğü gibidir.

```
1. $(seçici).fadeIn(hız);
2. $(seçici).fadeOut(hız);
3. $(seçici).fadeToggle(hız);
4. $(seçici).fadeTo(hız, saydamlık);
```

Şekil 12.15. “Fade” Metotlarının Sözdizimi

fadeIn() fonksiyonu

fadeIn() fonksiyonunun örnek uygulaması Şekil 12.16.’da görüldüğü gibidir.

Örnek uygulamada, Web sayfasında gizlenmiş şekilde bulunan kırmızı, yeşil ve mavi renkli kutular farklı efekt parametreleri ile görünür hâle gelmektedir.



jQuery’de efektlerin gerçekleşme hızını belirlerken “fast” (hızlı) ve “slow” (yavaş) anahtar kelimelerini de kullanabilirsiniz.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("button").click(function() {
8.     $("#div1").fadeIn();
9.     $("#div2").fadeIn("slow");
10.    $("#div3").fadeIn(3000);
11.   });
12. });
13. </script>
14. </head>
15. <body>
16. <h1>fadeIn fonksiyonu örnek uygulaması</h1>
17. <button>Kutuları görüntülemek için
   tıklayınız!</button><br><br>
18. <div id="div1"
   style="width:80px;height:80px;display:none;background-
   color:red;"></div><br>
19. <div id="div2"
   style="width:80px;height:80px;display:none;background-
   color:green;"></div><br>
20. <div id="div3"
   style="width:80px;height:80px;display:none;background-
   color:blue;"></div>
21. </body>
22. </html>
```

Şekil 12.16. Fadein() Fonksiyonu Örnek Uygulaması.

fadeOut() fonksiyonu

fadeOut() fonksiyonunun örnek uygulaması Şekil 12.17.’de görüldüğü

gibidir. Örnek uygulamada, Web sayfasında yer alan kırmızı, yeşil ve mavi renkli kutular farklı efekt parametreleri ile gizlenmektedirler.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("button").click(function() {
8.     $("#div1").fadeOut();
9.     $("#div2").fadeOut("slow");
10.    $("#div3").fadeOut(3000);
11.   });
12. });
13. </script>
14. </head>
15. <body>
16. <h1>fadeOut() fonksiyonu örnek uygulaması.</h1>
17. <button>Kutuları gizlemek için
   tıklayınız!</button><br><br>
18. <div id="div1"
   style="width:80px;height:80px;background-
   color:red;"></div><br>
19. <div id="div2"
   style="width:80px;height:80px;background-
   color:green;"></div><br>
20. <div id="div3"
   style="width:80px;height:80px;background-
   color:blue;"></div>
21. </body>
22. </html>

```

Şekil 12.17. Fadeout() Fonksiyonu Örnek Uygulaması.

fadeToggle() fonksiyonu

fadeToggle() fonksiyonunun örnek uygulaması Şekil 12.18.'de görüldüğü gibidir. Örnek uygulamada, butona ilk tıkladığında Web sayfasında yer alan kırmızı, yeşil ve mavi renkli kutular gizlenmekte, ikinci kere tıkladığında ise kutular görüntülenmektedir.



jQuery efektleri tüm HTML elemanlarına uygulanabilir.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("button").click(function() {
8.     $("#div1").fadeToggle();
9.     $("#div2").fadeToggle("slow");
10.    $("#div3").fadeToggle(3000);
11.   });
12. });
13. </script>
14. </head>
15. <body>
16. <h1>fadeToggle() fonksiyonu örnek uygulaması</h1>

```

```

17. <button>Kutuları gizleyip, göstermek için
    tıklayınız!</button><br><br>
18. <div id="div1"
    style="width:80px;height:80px;background-
    color:red;"></div>
19. <br>
20. <div id="div2"
    style="width:80px;height:80px;background-
    color:green;"></div>
21. <br>
22. <div id="div3"
    style="width:80px;height:80px;background-
    color:blue;"></div>
23. </body>
24. </html>

```

Şekil 12.18. Fadetoggle() Fonksiyonu Örnek Uygulaması.

fadeTo() fonksiyonu

fadeTo() fonksiyonunun örnek uygulaması Şekil 12.19.'da görüldüğü gibidir. Bu çalışmada, butona tıklandığında Web sayfasında yer alan kırmızı, yeşil ve mavi renkli kutulara farklı saydamlık parametreleri uygulanmaktadır.



Saydamlık (arka planı gösterme) jQuery efektlerin içinde kullanılabilir parametrelerden biridir.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("button").click(function() {
8.     $("#div1").fadeTo("slow", 0.15);
9.     $("#div2").fadeTo("slow", 0.4);
10.    $("#div3").fadeTo("slow", 0.7);
11.   });
12. });
13. </script>
14. </head>
15. <body>
16. <h1>fadeTo() fonksiyonu örnek uygulaması.</h1>
17. <button>Kutuları saydam hale getir!</button><br><br>
18. <div id="div1"
    style="width:80px;height:80px;background-
    color:red;"></div><br>
19. <div id="div2"
    style="width:80px;height:80px;background-
    color:green;"></div><br>
20. <div id="div3"
    style="width:80px;height:80px;background-
    color:blue;"></div>
21. </body>
22. </html>

```

Şekil 12.19. Fadeto() Fonksiyonu Örnek Uygulaması.

Şekil 12.19.'un 8., 9. ve 10. satırlarında ondalık değerler saydamlığı belirleyen numerik değerlerdir. Değer 1'e yaklaştıkça saydamlık oranı düşer. Örnekteki değerler göz önüne alındığında, kırmızı kutu saydamlığı en yüksek olan kutudur.



Bireysel Etkinlik

- Proje klasörünüzün içinde "soldurma.html" adlı bir dosya oluşturun.
- Şekil 12.19.'da yer alan kod yapısını örnek alarak sayfada oluşturduğunuz kutuların saydamlık değerlerini bir buton yardımıyla değiştirmeye çalışın.

Kayma (Slide)



jQuery efektleri içerisinde kullanılan "Toggle" metodu yardımıyla tek bir buton ile iki farklı olay gerçekleştirilebilir.

jQuery ile HTML elemanlarını "slide" metodunu kullanarak kaydırabilirsiniz. jQuery içinde "slide" metodunun farklı kullanım şekilleri yer almaktadır. Bunlar;

- **slideDown():** Aşağıya doğru kaydırma,
- **slideUp():** Yukarı doğru kaydırma,
- **slideToggle():** Aşağıya daha sonra ise yukarıya kaydırma (ya da tam tersi).

"slide" metotlarının sözdizimleri Şekil 12.20.'de görüldüğü gibidir.

```
1. $(seçici).slideDown(hız);
2. $(seçici).slideUp(hız);
3. $(seçici).slideToggle(hız);
```

Şekil 12.20. "Slide" Metotlarının Sözdizimi

slideDown() fonksiyonu

slideDown() fonksiyonunun örnek uygulaması Şekil 12.21.'de görüldüğü gibidir. Örnek uygulamada, HTML elemanı olarak oluşturulmuş olan bir panel aşağıya doğru kayarak açılmaktadır.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("#flip").click(function() {
8.     $("#panel").slideDown("slow");
9.   });
10. });
11. </script>
12. <style>
13. #panel, #flip {
14.   padding: 5px;
15.   text-align: center;
16.   background-color: #e5eccc;
17.   border: solid 1px #c3c3c3;
18. }
19. #panel {
20.   padding: 50px;
21.   display: none;
22. }
23. </style>
```

```

24. </head>
25. <body>
26. <div id="flip">Panel'i açmak için tıklayınız!</div>
27. <div id="panel">jQuery kayan panel örneği...</div>
28. </body>
29. </html>

```

Şekil 12.21. Slidedown() Fonksiyonu Örnek Uygulaması.

Şekil 12.21.'deki örnek uygulamanın 8. satırında panelin açılış hızını belirleyen "slow" değeri yer almaktadır. Bu değeri "fast" olarak değiştirerek panelin hızlıca açılmasını sağlayabilirsiniz.

slideUp() fonksiyonu

slideUp() fonksiyonunun örnek uygulaması Şekil 12.22.'de görüldüğü gibidir. Örnek uygulamada, HTML elemanı olarak oluşturulmuş olan bir panel yukarıya doğru kayarak kapanmaktadır.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function(){
7.   $("#flip").click(function(){
8.     $("#panel").slideUp("slow");
9.   });
10. });
11. </script>
12. <style>
13. #panel, #flip {
14.   padding: 5px;
15.   text-align: center;
16.   background-color: #e5eccc;
17.   border: solid 1px #c3c3c3;
18. }
19. #panel {
20.   padding: 50px;
21. }
22. </style>
23. </head>
24. <body>
25. <div id="flip">Panel'i kapatmak için
    tıklayınız!</div>
26. <div id="panel">jQuery kayan panel örneği...</div>
27. </body>
28. </html>

```

Şekil 12.22. Slidedown() Fonksiyonu Örnek Uygulaması.

Örnek uygulamanın yer aldığı Şekil 12.22.'nin 12. ve 22. satırları arasında yer alan bölüm CSS ile panelin tanımlandığı bölümdür. Bu bölüm üzerinde değişiklikler yapılarak farklı renk ve boyutta paneller elde edilebilir.



jQuery efektlerinin daha hızlı gerçekleşmesini istiyorsanız kod içindeki "slow" değerini "fast" ile değiştirebilirsiniz.

slideToggle() fonksiyonu

slideToggle() fonksiyonunun örnek uygulaması Şekil 12.23.'te görüldüğü gibidir. Örnek uygulamada, butona ilk tıkladığında kayarak açılan panel butona tekrar tıkladığında kayarak kapanmaktadır.



Farklı HTML elemanları ile çalışmak için ilgili elemanın CSS özelliklerini değiştirebilirsiniz.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("#flip").click(function() {
8.     $("#panel").slideToggle("slow");
9.   });
10. });
11. </script>
12. <style>
13. #panel, #flip {
14.   padding: 5px;
15.   text-align: center;
16.   background-color: #e5eccc;
17.   border: solid 1px #c3c3c3;
18. }
19. #panel {
20.   padding: 50px;
21.   display: none;
22. }
23. </style>
24. </head>
25. <body>
26. <div id="flip">Paneli açmak ya da kapatmak için
   tıklayınız!</div>
27. <div id="panel">jQuery kayan panel örneği...</div>
28. </body>
29. </html>
```

Şekil 12.23. Slidetoggle() Fonksiyonu Örnek Uygulaması.

jQuery'deki "Toggle" fonksiyonunu birçok farklı metot içinde inceleme fırsatı bulduk. Şekil 12.23.'teki örneğimizde de aynı HTML elemanı iki farklı davranışı "Toggle" kullanımı ile tetikleyebiliyor.



Bireysel
Etkinlik

- Proje klasörünüzün içinde "panel.html" adlı bir dosya oluşturun.
- Şekil 12.23.'te yer alan kod yapısını örnek alarak sayfa içerisinde açılıp kapanabilen bir bilgi paneli oluşturmaya çalışın.

Hareketlendirme (Animate)

jQuery ile HTML elemanlarına kendi belirlediğiniz şekilde (önceden tanımlanmış efektlerden bağımsız olarak) farklı hareket ve efektler vererek animasyonlar oluşturabilirsiniz. Bu animasyonlar HTML elemanlarının sayfa

üzerindeki konumlarını değiştirerek, renk ve boyutları üzerinde değişiklikler yaparak gerçekleştirilmektedir. jQuery'deki hareketlendirme uygulamalarını daha iyi anlayabilmek için animate() fonksiyonunun işleyişini genel hatları ile inceleyelim. animate() fonksiyonunun sözdizimi Şekil 12.24.'teki gibidir.

```
1. $(seçici).animate({parametreler},hız);
```

Şekil 12.24. Animate() Fonksiyonu Sözdizimi.

Şekil 12.24.'teki sözdizimini bir örnek uygulamada kullanalım. Örnek uygulamamızda Web sayfasında yer alan yeşil renkli bir kutu bulunduğu konumdan itibaren 250 piksel sağa doğru hareket etsin. Bu çalışmadaki kod yapısı Şekil 12.25.'teki gibi olmalıdır.



jQuery animate() fonksiyonu ile kendi belirlediğiniz animasyonları oluşturabilirsiniz.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("button").click(function() {
8.     $("div").animate({left: '250px'});
9.   });
10. });
11. </script>
12. </head>
13. <body>
14. <button>Animasyona başla!</button>
15. <p>Varsayılan olarak tüm HTML elemanları sayfa
    içerisinde statik bir konumdadır. CSS'in konum özelliği
    sayesinde sayfa içindeki konumları değişebilir.</p>
16. <div
    style="background:#98bf21;height:100px;width:100px;posi
    tion:absolute;"></div>
17. </body>
18. </html>
```

Şekil 12.25. Animate() Fonksiyonu Örnek Uygulama.

Şekil 12.25.'teki örnek uygulamanın 8. satırında animate() fonksiyonu içerisinde bazı parametrelerin tanımlandığını görmüşsünüzdür. Bu parametreler CSS'in konum özellikleri kullanarak HTML elemanının konumunu değiştirmektedir. Yeşil karenin sağa doğru hareket etmesine rağmen neden "left" yani sol olarak parametre tanımlandığı sorulabilir. Bunun cevabı soldan itibaren 250 piksel hareket etmesi istendiği içindir. CSS içerisindeki tanımlamalar bu şekilde yapılır.



jQuery animate() fonksiyonu ile HTML elemanlarının birden çok özelliği değiştirilebilir.

jQuery'deki animasyon özelliği sadece tek bir özelliğin değiştirilmesi ile sınırlı değildir. HTML elemanının birden çok özelliği de değiştirilebilir ve animasyon daha karmaşık bir yapıya kavuşabilir. Şekil 12.26.'da önceki örneğimizde kullandığımız yeşil kare bu sefer sağa doğru 300 piksel hareket etmekte ve boyutlar ile saydamlık değeri de değişiklik göstermektedir.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.     $("button").click(function() {
8.         $("div").animate({
9.             left: '250px',
10.             opacity: '0.5',
11.             height: '150px',
12.             width: '150px'
13.         });
14.     });
15. });
16. </script>
17. </head>
18. <body>
19. <button>Animasyona başla!</button>
20. <p>Varsayılan olarak tüm HTML elemanları sayfa
    içerisinde statik bir konumdadır ve boyut ve renk gibi
    özellikleri de sabittir. CSS sayesinde belirlenen
    özellikler sayfa içinde değişebilir.</p>
21. <div
    style="background:#98bf21;height:100px;width:100px;posi
    tion:absolute;"></div>
22. </body>
23. </html>

```

Şekil 12.26. Animate() Fonksiyonu İle Birden Fazla Özelliğin Değiştirilmesi.

animate() fonksiyonunu kullandığımız örneklerimizde (Şekil 12.24., 16. satır, Şekil 12.26., 21. satır) HTML elemanının (yeşil kutunun) ilk tanımlaması <div> etiketleri yardımıyla yapılır. Bu tanımlamaları istediğiniz gibi değiştirerek farklı HTML elemanları oluşturabilirsiniz.

jQuery'deki animasyon kullanımının ilginç noktalarından biri de HTML elemanının CSS özelliklerini değiştirirken verilen değer sabit değil, bir önceki değere bağlı olarak verilebilmesidir. Bu özelliği gözlemleyebileceğimiz örneğimizin kodları Şekil 12.27.'de yer almaktadır. Örnek uygulamada sayfada yer alan kırmızı bir kutu, butona her tıklandığında 50 piksel sağa doğru hareket edecektir.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.     $("button").click(function() {
8.         $("div").animate({
9.             left: '+=50px'
10.         });
11.     });
12. });
13. </script>
14. </head>
15. <body>

```

```

16. <button>Animasyona başla!</button>
17. <p>Bir önceki konumuna göre hareketini güncelleyen
    kutu örneği.</p>
18. <div
    style="background:#FF0000;height:100px;width:100px;posi
    tion:absolute;"></div>
19. </body>
20. </html>

```

Şekil 12.27. Önceki Değere Bağlı Olarak Animasyonun Gerçekleşmesi.

Şekil 12.27.'deki örneğin 9. satırında parametre tanımlaması yapılırken statik bir değer belirlemek yerine “+=” ifadesi kullanılarak bir önceki konuma 50 piksel eklenmesi sağlanmaktadır. Bu sayede **butona her tıklandığında** kırmızı kutunun mevcut konumu değişmektedir.

jQuery'deki animate() fonksiyonu tek bir tetikleme ile birden çok animasyonun arka arkaya gerçekleştirilmesine de imkân sağlar. Şekil 12.28.'de mavi bir kutunun hem boyut, hem konum, hem renk, hem de saydamlık değerleri belirli bir sırada gerçekleştirilmektedir.



Tek bir fare hareketi ile
birden çok animasyon
tetiklenebilir.

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.   $("button").click(function() {
8.     var div = $("div");
9.     div.animate({left: '200'}, "fast");
10.    div.animate({height: '300px', opacity: '0.4'},
11.      "slow");
12.    div.animate({width: '300px', opacity: '0.8'},
13.      "slow");
14.    div.animate({height: '100px', opacity: '0.4'},
15.      "slow");
16.    div.animate({width: '100px', opacity: '0.8'},
17.      "slow");
18.    div.animate({left: '10'}, "fast");
19.  });
20. });
21. </script>
22. </head>
23. <body>
24. <button>Animasyona başla!</button>
25. <p>Animasyonların belirli bir sırada
    gerçekleştirilmesi.</p>
26. <div
    style="background:#0000FF;height:100px;width:100px;posi
    tion:absolute;"></div>
27. </body>
28. </html>

```

Şekil 12.28. Animasyonların Belirli Bir Sırada Gerçekleştirilmesi.



Bireysel Etkinlik

- Proje klasörünüzün içinde "animasyon.html" adlı bir dosya oluşturun.
- Sayfada oluşturduğunuz bir HTML elemanının konumunu bir buton yardımıyla değiştirmeye çalışın. Butona her tıkladığınızda ilgili eleman ekranda 25 piksellik bir hareket gerçekleştirsin.

Durdurma (Stop)

jQuery'deki stop() fonksiyonu uygulama içerisindeki animasyonların henüz bitmeden önce durdurulması için kullanılır. stop() fonksiyonunun sözdizimi Şekil 12.29.'daki gibidir.



jQuery'de animasyonlar normal sürelerinden önce durdurulabilirler.

```
1. $(seçici).stop(tümünüDurdur,sonaGit);
```

Şekil 12.29. Stop() Fonksiyonu Sözdizimi.

jQuery'de stop() fonksiyonu iki farklı şekilde kullanılmaktadır. Bu kullanım şekilleri aşağıda listelenmiştir:

- **Parametresiz kullanım:** stop() fonksiyonu bu kullanımda herhangi bir parametre almaz. O an hangi animasyon gerçekleştiriliyorsa onu durdurur. Bir sonraki animasyonun başlamasını engellemez. Bir sonraki animasyonun durdurulması için ikinci bir tetiklemeye ihtiyaç duyulur.
- **Parametrelili kullanım:** stop() fonksiyonu Şekil 12.29.'da gösterildiği gibi iki farklı parametre alabilir, bu parametrelerden ilki tüm animasyonların durdurulması ile ilgili parametredir. Tetikleme gerçekleştiği anda animasyon durdurulur ve eğer bir sonraki animasyon varsa bile o animasyona geçiş yapmaz. İkinci parametre ise animasyonu durdurur ve o animasyonun son karesine gider. Parametreleri aktif hâle getirmek için "true" (doğru) değerine fonksiyon içinde yer verilir.

stop() fonksiyonunun parametrelili ve parametresiz kullanımı Şekil 12.30.'daki örnek uygulamada yer almaktadır.



stop() fonksiyonu iki farklı parametre alabilir.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="jquery.js"></script>
5. <script>
6. $(document).ready(function() {
7.     $("#start").click(function() {
8.         $("div").animate({left: '100px'}, 5000);
9.         $("div").animate({fontSize: '2em'}, 5000);
10.    });
11.
12.    $("#stop").click(function() {
13.        $("div").stop();
14.    });
15.
16.    $("#stop2").click(function() {
```

```
17.     $("div").stop(true);
18.     });
19.
20.     $("#stop3").click(function(){
21.         $("div").stop(true, true);
22.     });
23. });
24. </script>
25. </head>
26. <body>
27. <button id="start">Animasyona başla!</button>
28. <button id="stop">Animasyonu durdur!</button>
29. <button id="stop2">Tüm animasyonları durdur!</button>
30. <button id="stop3">Durdur ve sona git</button>
31. <p>stop() animasyonunun parametrelili ve parametresiz
    kullanımı.</p>
32. <div
    style="background:#00FF00;height:100px;width:200px;position:absolute;">Dur fonksiyonu</div>
33. </body>
34. </html>
```

Şekil 12.30 Stop() Fonksiyonu Örnek Uygulama.



Özet

- jQuery, hızlı, küçük ve zengin özelliklere sahip bir JavaScript kütüphanesidir.
- jQuery, platform bağımsız olarak tanımlayabileceğimiz bir yapıda tasarlanmıştır. Bu yapısı birçok farklı İnternet tarayıcısında sorunsuz olarak çalışabilmesini sağlar.
- HTML belgesinde gezinme, olay işleme, animasyon ve diğer Ajax uygulamalarını çok daha basit hâle getirir.
- jQuery, açık kaynak kodlu bir projedir.
- JavaScript kütüphanelerinin genel çalışma prensipleri jQuery için de geçerlidir.
- jQuery kütüphane dosyasına iki farklı şekilde bağlantı sağlanabilir; dosya yerel bir adreste bulunabilir, İnternet üzerinde yer alan depolar üzerinden kullanılabilir.
- jQuery, kod geliştiriciler ve Web tasarımcılar için farklı kütüphane sürümleri sunar; bu sürümler arasındaki fark kodlar üzerinde değişiklik yapma kolaylığı sağlaması ile ilgilidir.
- jQuery sözdizimi şablonu temel olarak iki bölümden oluşur; seçici ve eylem. Seçici kısmı işlem yapılacak HTML elemanının seçilmesini, eylem kısmı ise gerçekleştirilecek fonksiyonu belirlememizi sağlar.
- jQuery Web sayfasının tamamı yüklenmeden çalışmaya başlaması durumunda hataya neden olabilir. Bu hatayı engellemek için "doküman hazır olayı" adlı özel bir fonksiyon yer almaktadır.
- jQuery'de bir fonksiyonun çalışması için farklı tetikleme seçenekleri bulunmaktadır. Bu seçenekler; fare olayları, klavye olayları, form olayları ve doküman olaylarıdır.
- jQuery içerisinde kullanıcı etkileşimini arttıracak ve kullanıcı deneyimini daha eğlenceli hale getirebilecek efektler yer almaktadır. jQuery efektleri teker teker ya da bir arada kullanılabilirdiği gibi, birden fazla efekt belirli bir sırayla da kullanılabilir.
- Gizle / Göster efekti: jQuery ile HTML elemanlarını gizle (hide ve göster (show) metotları ile gizleyebilir ya da görünür hale getirebilirsiniz.
- Soldurma efekti: jQuery ile HTML elemanlarını soldurma (fade) metodunu kullanarak soldurma efekti ile gizleyebilir ya da belirginleştirme efekti ile görünür hale getirebilirsiniz.
- Kayma efekti: jQuery ile HTML elemanlarını kayma (slide) metodunu kullanarak kaydırabilirsiniz.
- Hareketlendirme efekti: jQuery ile HTML elemanlarına kendi belirlediğiniz şekilde (önceden tanımlanmış efektlerden bağımsız olarak) farklı hareket ve efektler vererek animasyonlar oluşturabilirsiniz. Bu animasyonlar HTML elemanlarının sayfa üzerindeki konumlarını değiştirerek, renk ve boyutları üzerinde değişiklikler yaparak gerçekleştirilmektedir.
- Hareketlendirme efekti uygulanmış çalışmalarda süreç tamamlanmadan animasyon durdurulmak isteniyorsa dur (stop) metodu kullanılabilir.

DEĞERLENDİRME SORULARI

1. HTML standartlarının ilk kullanılmaya başlandığı yıl aşağıdakilerden hangisidir?
 - a) 1991
 - b) 1993
 - c) 1998
 - d) 2003
 - e) 2005
2. Aşağıdakilerden hangisi HTML'in tek başına (herhangi bir ek bileşen olmadan) sahip olduğu özelliklerden biri değildir?
 - a) Farklı İnternet tarayıcıları üzerinde çalışabilir.
 - b) Metin tabanlı bir etiketleme dili kullanılır.
 - c) Sayfa içerisindeki veri güncellemelerinde sayfanın yenilenmesine (refresh) gerek yoktur.
 - d) HTML özel olarak derlenmesine gerek kalmadan İnternet tarayıcılarda yorumlanabilir.
 - e) HTML kaynak kodları herhangi bir metin editöründe görüntülenebilir.
3. jQuery kütüphane dosyalarının sıkıştırılmış sürümlerinin (min) sağladığı avantaj aşağıdakilerden hangisidir?
 - a) Boyut olarak daha küçük oldukları için Web sitesi optimizasyonu sağlar.
 - b) Dosyaların içeriğini herkes göremez, koruma sağlar.
 - c) Güncelleme işlemleri daha kolay gerçekleştirilir.
 - d) Farklı sunuculara aktarılması kolaydır.
 - e) Daha çok fonksiyonu içinde barındırabilir.
4. İçerik Dağıtım Ağı (Content Delivery Network – CDN) ne işe yarar?
 - a) Aynı içeriğin farklı Web sitelerinde yayınlanabilmesini sağlar.
 - b) Web sitelerindeki görsel ve işitsel materyalleri depolar.
 - c) Günlük gelişmelerin İnternet üzerinden takip edilebileceği bir platform sunar.
 - d) İhtiyaç duyulan kaynak kodların İnternet üzerinde yer alan depolardan kullanılabilmesini sağlar.
 - e) jQuery'de oluşturulmuş içeriklerin yedeklenmesini sağlar.

5. Doküman hazır olayı (The Document Ready Event) jQuery ile geliştirilen uygulamalar açısından neden önemlidir?
 - a) Dokümanın İnternet tarayıcıda daha hızlı yüklenmesini sağlar.
 - b) Yeni bir uygulama geliştirirken hazır olan jQuery kodlarını kullanmamızı sağlar.
 - c) İki farklı jQuery olayı arasında bağlantı kurar.
 - d) Web sitesinin jQuery ile uyumlu olup olmadığını kontrol eder.
 - e) Dokümanın yüklenmesi henüz tamamlanmadan jQuery kodlarının çalışmasına engel olarak olası bir hatayı engeller.
6. jQuery seçicileri Web sayfası içerisindeki hangi tür öğelerin seçiminde kullanılır?
 - a) JavaScript kodları içindeki fonksiyonların
 - b) HTML elemanlarının
 - c) Jpg, gif, png gibi görsel elemanların
 - d) Mp4, avi gibi akışkan medyanın
 - e) Mp3, wav gibi işitsel elemanların
7. Aşağıdakilerden hangisi jQuery fare olaylarından biri değildir?
 - a) Tıklama
 - b) Çift tıklama
 - c) Fare tekerleğinin döndürülmesi
 - d) Fare girişi
 - e) Fare çıkışı
8. Aşağıdakilerden hangisi jQuery efektleri içerisinde diğerlerine göre farklı bir işleve sahiptir?
 - a) Gizle / Göster
 - b) Soldurma
 - c) Kaydırma
 - d) Hareketlendirme
 - e) Durdurma
9. jQuery efektlerinde gerçekleştirilen işlemin aynı tetikleyici yardımıyla geri alınabilmesini sağlayan metot aşağıdakilerden hangisidir?
 - a) Show
 - b) Fade
 - c) Stop
 - d) Toggle
 - e) Slide

10. Açık kaynak kodlu bir proje olan jQuery hangi lisans şartları altında dağıtılmaktadır?

- a) MIT
- b) GNU
- c) Creative Commons
- d) Telif Hakları
- e) Apache

Cevap Anahtarı

1.b, 2.c, 3.a, 4.d, 5.e, 6.b, 7.c, 8.e, 9.d, 10.a

YARARLANILAN KAYNAKLAR

Baltalı, S. (2011). jQuery. İstanbul: KODLAB

jQuery Öğrenme Merkezi. 19 Haziran 2019 tarihinde <https://learn.jquery.com/> adresinden erişildi.

jQuery Resmî Web Sitesi. 17 Haziran 2019 tarihinde <https://jquery.com/> adresinden erişildi.

Raggett, D., Lam, J., Alexander, I. ve Kmieć, M. (1999). History of HTML. Essex, England: Addison Wesley Longman Limited. 17 Haziran 2019 tarihinde <https://www.w3.org/People/Raggett/book4/ch02.html> adresinden erişildi.

Software Studio. MIT Courseware. 12 Temmuz 2019 tarihinde https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-170-software-studio-spring-2013/recitations/MIT6_170S13_rec6-jQuery.pdf adresinden erişildi.

W3Schools. jQuery Ders Notları. 3 Haziran 2019 tarihinde <https://www.w3schools.com/jquery> adresinden erişildi.

Web Programlama 302: jQuery. Turkcell Geleceği Yazarlar. 22 Haziran 2019 tarihinde <https://gelecegiyazarlar.turkcell.com.tr/konu/web-programlama/egitim/302> adresinden erişildi.