



CodeCraft

Bootcamp'25

jQuery Effects & jQuery Plugins & Utilities

3. Gün

Konu Başlıkları

- **jQuery Animate Syntax**
- **jQuery Animate Özellikleri**
- **jQuery Effects Hide Show Toggle**
- **jQuery Effects Fading**
- **jQuery Effects Fade Metodları**
- **jQuery Sliding Efektleri**
- **jQuery Sliding Özellikleri**
- **jQuery Stop Animations**
- **jQuery Stop Animations Kullanımı**
- **jQuery Effects: Queue**
- **jQuery Plugins**
- **Popüler jQuery Eklentileri ve Kullanım Alanları**
- **jQuery Validate Plugin Nedir?**
- **jQuery Validate Plugin'i Nasıl Kullanırırmı?**
- **jQuery Validate Plugin Parametreleri**
- **jQuery Plugin Oluşturma**
- **jQuery Utilities (Yardımcı Araçlar)**

jQuery Animate Syntax

jQuery, `.animate()` metodu ile CSS özelliklerini zaman içinde değiştirebilir, kullanıcı deneyimini zenginleştiren dinamik görseller oluşturabilirsiniz. Animasyonların süresini ve hız eğrisini belirleyebilir, geri çağrıma fonksiyonları ile etkileşimleri özelleştirebilirsiniz.

```
$(selector).animate({params}, speed, callback);
```

Bileşen	Açıklama	Örnek Kullanım
selector	Animasyon uygulanacak öğeyi belirler.	<code>\$("#box").animate(...)</code>
{params}	CSS özelliklerini ve yeni değerlerini belirtir.	<code>{ width: "300px", opacity: 0.5 }</code>
speed (opsiyonel)	Animasyonun süresini belirler (milisaniye veya "slow", "fast" gibi hazır değerler kullanılabilir).	<code>"slow", 1000 (1 saniye)</code>
callback (opsiyonel)	Animasyon tamamlandıktan sonra çalışacak fonksiyon.	<code>function() { alert("Bitti!"); }</code>

jQuery Animate Özellikleri

Özellik	Açıklama	Örnek
Birden Fazla Özelliğin Birlikte Değiştirilmesi	Aynı anda birden fazla CSS özelliğini değiştirebilirsiniz. Bu, bir öğenin birkaç özelliğini aynı anda animasyonla değiştirmenizi sağlar.	<code>\$("#box").animate({ width: "300px", opacity: 0.5 }, 1000)</code>
Göreli Değerler Kullanma	Mevcut konumuna göre görelî (relative) değerler kullanarak animasyon yapılabılır. Öğenin mevcut pozisyonuna göre hareket etmesini sağlar.	<code>\$("#box").animate({ left: "+=50px", top: "-=50px" }, 1000);</code>
Önceden Tanımlanmış Değerler Kullanma	jQuery'nin önceden tanımladığı animasyon fonksiyonlarını kullanarak basit geçişler yapılabilir. fadeIn(), fadeOut(), show(), hide() gibi fonksiyonlar içerir.	<code>\$("#box").fadeOut(1000);</code>
Kuyruk Fonksiyonu Kullanma	Animasyonlar, sırayla çalışacak şekilde kuyruk işleviyle sıralanabilir. Bir animasyon tamamlandığında bir sonrakine geçilir.	<code>\$("#box").animate({ left: "100px" }, 1000).animate({ top: "100px" }, 1000);</code>

jQuery Effects Hide Show Toggle

jQuery, öğeleri gizlemek ve göstermek için basit ve etkili yöntemler sunar. `hide()` ve `show()` fonksiyonları öğelerin görünürlüğünü değiştirirken, `toggle()` fonksiyonu ise görünürlüğünü tersine çevirir. Yani bir öğe görünürse gizler, gizliyse gösterir. Bu, dinamik kullanıcı etkileşimleri sağlar.

```
$(selector).show/hide/toggle(speed,callback);
```

İfade	Açıklama
selector	Seçilen öğe veya öğeler. jQuery seçicisi (<code>selector</code>) kullanılarak, animasyonun uygulanacağı HTML ögesi belirtilir.
speed	Animasyonun süresi. Bu, animasyonun hızını belirler. Genellikle milisaniye cinsinden bir sayı veya önceden tanımlı slow, fast değerleri kullanılır.
callback	Animasyon tamamlandığında çalıştırılacak fonksiyon. Bu isteğe bağlı bir parametre olup, animasyon bitiminde çalışacak bir geri çağrıma fonksiyonu belirtir.

jQuery Effects Fading

jQuery'nin **fading** efektleri, öğelerin opaklık (opacity) seviyelerini değiştirmek için kullanılır. Bu, öğelerin yavaşça görünür veya gizli hale gelmesini sağlar. jQuery'de en yaygın kullanılan fading fonksiyonları `fadeIn()`, `fadeOut()`, ve `fadeToggle()`'dur.

Syntax

```
$(selector).fadeIn/fadeOut/fadeToggle(speed,callback);
```

```
$(selector).fadeTo(speed,opacity,callback);
```

jQuery Effects Fade Metodları

Fonksiyon	Açıklama
fadeIn()	Gizli bir öğeyi yavaşça görünür hale getirir. Öğenin opaklısı sıfırdan 1'e doğru artar.
fadeOut()	Görünür bir öğeyi yavaşça gizler. Öğenin opaklısı 1'den sıfıra doğru azalır.
fadeToggle()	Öğenin görünürlüğünü tersine çevirir. Eğer öğe görünüyorrsa gizler, gizliyse gösterir.
fadeTo()	Bir öğenin opaklı değerini belirtilen seviyeye yavaşça değiştirir. Bu, öğenin tamamen görünür olmasından yarı saydam olmasına kadar her değeri alabilir.

jQuery Sliding Efektleri

jQuery'nin **sliding** efektleri, öğelerin yatay (sol/sağ) veya dikey (aşağı/yukarı) yönde kaymasını sağlar. Bu efektler, öğeleri kaydırarak gizlemek, göstermek veya boyutlandırmak için kullanılır. `slideUp()`, `slideDown()`, ve `slideToggle()` gibi fonksiyonlar, bu tür kaydırma efektlerini gerçekleştirmek için en yaygın kullanılan fonksiyonlardır.

Syntax

```
$(selector).slideDown/slideUp/slideToggle(speed,callback);
```

jQuery Sliding Özellikleri

Fonksiyon	Açıklama
slideDown()	Gizli bir öğeyi aşağı kaydırarak gösterir. Ögenin yüksekliği sıfırdan normal boyutuna doğru açılır.
slideUp()	Görünür bir öğeyi yukarı kaydırarak gizler. Ögenin yüksekliği normal boyutundan sıfıra iner.
slideToggle()	Ögenin görünürlüğünü tersine çevirir. Eğer öğe görünüyorrsa yukarı kaydırarak gizler, gizliyse aşağı kaydırarak gösterir.

jQuery Stop Animations

jQuery'deki **stop()** fonksiyonu, animasyonları durdurmak ve animasyon kuyruğunda bekleyen işlemleri sonlandırmak için kullanılır. Bu özellik, uzun süren animasyonlar veya birden fazla animasyonun çakışması durumunda oldukça faydalıdır. stop() fonksiyonu, animasyonları hemen durdurur veya animasyonu belirli bir süre sonra sonlandırmak için kullanılabilir.

Syntax

```
$(selector).stop(stopAll,goToEnd);
```

jQuery Stop Animations Kullanımı

Fonksiyon	Açıklama	Örnek
stop()	Animasyonu hemen durdurur, ögeyi son haline getirmez.	<code>\$(".intro").stop();</code>
stop(true)	Animasyonu hemen durdurur ve ögeyi animasyonun son haline getirir.	<code>\$(".intro").stop(true);</code>
stop(false)	Animasyonu durdurur ancak ögeyi animasyonun yarı kalmış durumunda bırakır.	<code>\$(".intro").stop(false);</code>
stop(true, true)	Animasyonu durdurur ve aynı anda animasyon kuyruğundaki tüm bekleyen işlemleri temizler.	<code>\$(".intro").stop(true, true);</code>

jQuery Effects: Queue

jQuery animasyon kuyruğunu göstermek ve manipüle etmek için **queue()** fonksiyonu kullanılabilir. jQuery, bir öğe üzerinde birden fazla animasyon işlemi olduğunda bu işlemleri sıraya koyar (queue). Ancak, doğrudan animasyon kuyruğunu görmek veya izlemek için jQuery'nin gömülü bir aracı yoktur. Yine de kuyruğu incelemek ve göstermek için bazı dolaylı yöntemler kullanabilirsiniz.

Metod	Açıklama	Örnek Kullanım
queue()	Kuyruğu alır ve animasyonları sırasıyla döndürür.	<code>\$(".box").queue();</code>
clearQueue()	Kuyruğu temizler ve tüm bekleyen animasyonları iptal eder.	<code>\$(".box").clearQueue();</code>
next()	Kuyruğa eklenen bir işlevi çalıştırmak için kullanılır ve sıradaki animasyonu başlatır.	<code>\$(".box").queue(function(next) { console.log("Animasyon devam ediyor"); next(); });</code>

jQuery Plugins

jQuery eklentileri, jQuery'nin sunduğu temel fonksiyonları genişleten ve belirli bir işlemi daha kolay hale getiren harici modüllerdir. Bir eklenti, tekrar eden işlemlerini basitleştirmek için kod yazılmasını sağlar. jQuery topluluğu tarafından geliştirilen binlerce eklenti, çeşitli kullanım alanlarına hitap eder. Örneğin, kaydırıcı (slider), lightbox, modal penceler, form doğrulama ve veri tabloları gibi bileşenleri kolayca eklemek için jQuery eklentileri kullanılabilir.

`$(selector).pluginName(options);`

Syntax	Açıklama
<code>\$(selector)</code>	jQuery ile seçilen HTML öğesini belirtir.
<code>.pluginName</code>	Kullanılacak eklentinin adıdır. (Örn: slick(), lightbox(), datepicker())
<code>(options)</code>	Eklentiye özelleştirilebilir ayarlar (opsiyonel) eklenir. Varsayılan değerler kullanılabilir.
Tam Örnek Kullanım	<code>\$(".slider").slick({autoplay: true, dots: true});</code>

Popüler jQuery Eklentileri ve Kullanım Alanları

Bu eklentiler sayesinde **kaydırıcılar (sliders)**, **modal pencereler**, **form doğrulama sistemleri**, **tablo yönetimi** gibi birçok fonksiyon ekleyebiliriz. Aşağıda, en popüler jQuery eklentileri ve kullanım alanları açıklanmaktadır:

Eklenti Adı	Kullanım Alanı	Örnek Kullanım
Slick	Kaydırıcı (Slider) bileşeni eklemek için kullanılır.	<code>\$(".slider").slick({ autoplay: true });</code>
Owl Carousel	Resim veya içerik kaydırıcıları oluşturmak için popülerdir.	<code>\$(".owl-carousel").owlCarousel({ loop: true });</code>
Magnific Popup	Hafif ve duyarlı modal açılır pencereler oluşturur.	<code>\$(".popup-link").magnificPopup();</code>
jQuery Validate	Form girişlerini doğrulamak için kullanılır.	<code>\$("#form").validate();</code>
DataTables	Büyük veri tablolarını dinamik olarak yönetmek için kullanılır.	<code>\$("#example").DataTable();</code>

jQuery Validate Plugin Nedir?

jQuery Validate Plugin, web formlarını doğrulamak için kullanılan bir JavaScript kütüphanesidir. Form doğrulama, kullanıcıların formları doğru şekilde doldurup doldurmadığını kontrol etmek için önemlidir. Bu kütüphane, form alanlarını otomatik olarak kontrol eder ve hatalı verileri kullanıcıya bildiren mesajlar gösterir. Böylece kullanıcı, hatalı bilgi girmeden önce hatalarını düzeltebilir.

Özellik	Açıklama
Basit ve Hızlı Kullanım	Kısa sürede form doğrulaması ekleyebilirsiniz. Sadece birkaç satır kod ile tüm form doğrulamanızı halledebilirsiniz.
Özel Hata Mesajları	Form alanları için özel hata mesajları yazabilir ve kullanıcıya anlamlı uyarılar verebilirsiniz.
Çoklu Doğrulama Kuralları	E-posta, telefon numarası, URL gibi farklı veri türleri için doğrulama kuralları oluşturabilirsiniz.
Dinamik Doğrulama	Form alanları, kullanıcı her değiştirdiğinde otomatik olarak doğrulanabilir.
Formu Göndermeyi Engelleme	Formdaki bir alan hatalıysa, formun gönderilmesini engeller. Kullanıcı hatalarını düzeltene kadar formu gönderemez.
Farklı Doğrulama Yöntemleri	Kendi doğrulama fonksiyonlarınızı yazabilir ve özel kurallar ekleyebilirsiniz.

jQuery Validate Plugin'i Nasıl Kullanırıım?

Adımlar	Açıklama
1. jQuery ve jQuery Validate Plugin'i Dahil Etme	İlk olarak, jQuery ve jQuery Validate Plugin'i sayfanıza dahil etmeniz gerekmektedir. jQuery, JavaScript işlevlerini kolayca kullanmanıza, Validate Plugin ise form doğrulamasını gerçekleştirmenize olanak tanır. Bu kütüphaneleri CDN üzerinden sayfanıza ekleyebilirsiniz.
2. HTML Formunu Oluşturma	Form, kullanıcıdan veri almak için gerekli alanları içerir. Form elemanları (ad, e-posta, telefon vb.) input etiketleriyle oluşturulur ve her bir alanın doğrulaması yapılacak şekilde id ve name özellikleri ile tanımlanır.
3. Formu Doğrulama Kurallarını Belirleme	Form alanları için doğrulama kuralları belirlenir. Bu kurallar, kullanıcı verilerinin doğru formatta olup olmadığını kontrol eder. Doğrulama, kullanıcı hatalı bir veri girerse uyarı mesajları gösterir ve formu gönderilmesini engeller.

jQuery Validate Plugin Parametreleri

```
$("document").ready(function() {
    $("#myForm").validate({
        rules: {
            name: { required: true, minlength: 2 },
            email: { required: true, email: true },
            phone: { required: true, minlength: 10 }
        },
        messages: {
            name: { required: "Ad gereklidir", minlength: "En az 2 karakter" },
            email: { required: "E-posta gereklidir", email: "Geçerli e-posta" },
            phone: { required: "Telefon gereklidir", minlength: "En az 10 hanelidir" }
        },
        submitHandler: function(form) {
            alert("Form gönderildi!");
            form.submit();
        },
        errorClass: "error",
        validClass: "valid",
        errorPlacement: function(error, element) { error.insertAfter(element); },
        focusInvalid: true
    });
});
```

Parametre	Açıklama
rules	Form alanları için doğrulama kurallarını belirtir. (Örneğin, required, email, minlength gibi)
messages	Her bir form alanı için özel hata mesajlarını tanımlar.
submitHandler	Form doğrulandıktan sonra yapılacak işlemi belirtir. Örneğin, formu gönderme işlemi.
errorClass	Hatalı form alanlarına uygulanan CSS sınıfını belirtir.
validClass	Geçerli form alanlarına uygulanan CSS sınıfını belirtir.
errorPlacement	Hata mesajlarının sayfada nereye yerleştirileceğini belirler.
focusInvalid	Hatalı bir form alanı olduğunda, bu alana otomatik olarak odaklanılmasını sağlar.

jQuery Plugin Oluşturma

Bir jQuery eklentisi yazarken, genellikle \$ işlevini genişleteerek \$.fn.pluginName yapısını kullanırız. Bu, jQuery'nin tüm ögeleriyle uyumlu bir şekilde çalışmasını sağlar. Aşağıda özel bir jQuery eklentisinin temel syntax yapısını açıklıyoruz:

```
(function($) {  
    $.fn.pluginName = function() {  
        return this.each(function() {  
            $(this).css("color", "red");  
        });  
    };  
})(jQuery);
```

Kod Parçası	Açıklama
(function(\$) { })}(jQuery);	\$ simgesini güvenli hale getirir ve eklentinin diğer kütüphanelerle çakışmasını önler.
\$.fn.pluginName = function(){ };	pluginName adında bir eklenti oluşturur. Kullanıcı, bu eklentiyi jQuery seçenekleriyle çağırabilir.
return this.each(function(){ });	Seçilen tüm ögeler için işlemi tekrarlar.
\$(this).css("color", "red");	Seçilen ögenin rengini kırmızı yapar.

jQuery Utilities (Yardımcı Araçlar)

jQuery Utilities (Yardımcı Araçlar), jQuery'nin sunduğu yardımcı fonksiyonlardır ve çeşitli işlemleri daha pratik hale getirir. Bu fonksiyonlar, nesne manipülasyonu, dizi işlemleri, veri yönetimi gibi konularda kullanılır.

En yaygın jQuery utility fonksiyonları:

Metod	Açıklama
<code>\$.each()</code>	Bir dizi veya nesne üzerindeki tüm elemanları döngüyle işler.
<code>\$.extend()</code>	Bir nesnenin özelliklerini başka bir nesneye kopyalar.
<code>\$.isArray()</code>	Verilen değişkenin bir dizi olup olmadığını kontrol eder.
<code>\$.trim()</code>	Bir string'in başındaki ve sonundaki boşlukları kaldırır.
<code>\$.type()</code>	Bir değişkenin veri türünü döndürür.
<code>\$.inArray()</code>	Belirtilen değerin bir dizide olup olmadığını kontrol eder.

Ödev

Hedef: Bir iş başvuru formu oluşturup, formu animasyonlarla gösterecek ve doğrulama işlemleri yapacaksınız. Ayrıca, bazı jQuery eklentilerini kullanarak form doğruluğunu artıracaksınız.

Gereklilıklar:

Bu projede öncelikle bir başvuru formu oluşturacağız. Formda **ad**, **soyad**, **e-posta**, **telefon numarası** ve **pozisyon** gibi alanlar olacak. Ayrıca bir de "**Gönder**" butonu ekleyeceğiz. Formu başlangıçta gizleyeceğiz ve kullanıcı "**Başvuru Yap**" butonuna tıkladığında formu **fadeIn()** ile göstereceğiz. Formu kapatmak istediğimizde **fadeOut()** kullanacağız.

Formu oluşturuktan sonra, form alanlarını doğrulamak için **jQuery Validate Plugin** kullanacağız. Bu sayede e-posta ve telefon numarasının doğru formatta olup olmadığını kontrol edeceğiz. Eğer bir alan boş veya hatalıysa, kullanıcıya hata mesajı göstereceğiz.

Form başarıyla gönderildikten sonra bir başarı mesajı göstereceğiz. Bu mesajı **fadeIn()** ile ekrana getireceğiz ve 3 saniye sonra **fadeOut()** ile gizleyeceğiz.

Son olarak, opsiyonel olarak bazı plugin'ler kullanabiliriz. Örneğin, tarih seçimi eklemek için **jQuery UI Datepicker** kullanabiliriz. Ayrıca telefon numarasına format eklemek için **jQuery Masked Input** kullanabiliriz.