



CodeCraft

Bootcamp'25

Advanced DOM Manipulation

4. Gün

Konu Başlıklarısı

- DOM Manipulation Nedir?
- Traversing - DOM Ağacında Gezinme
- jQuery Traversing - Ancestors (Üst Öğeler)
- jQuery Traversing - Descendants (Alt Öğeler)
- jQuery Traversing - Siblings (Kardeş Öğeler)
- jQuery Traversing - Filtering (Öğeleri Filtreleme)
- jQuery Traversing - Tüm Metodlar
- Cloning - Elemanları Kopyalama
- Cloning Kullanım Senaryoları

DOM Manipulation Nedir?

Web sayfaları, HTML ve CSS ile oluşturulur ancak sayfalar genellikle statik başlar. Kullanıcı etkileşimi, animasyonlar, veri güncellemleri ve dinamik içerikler oluşturmak için **DOM manipülasyonu** kullanılır. DOM (Document Object Model), bir web sayfasındaki tüm HTML öğelerini bir ağaç yapısı şeklinde temsil eder. jQuery ile DOM üzerinde hızlı ve etkili değişiklikler yapabiliriz.

Bu derste, **gelişmiş DOM manipülasyonu** tekniklerini inceleyeceğiz:

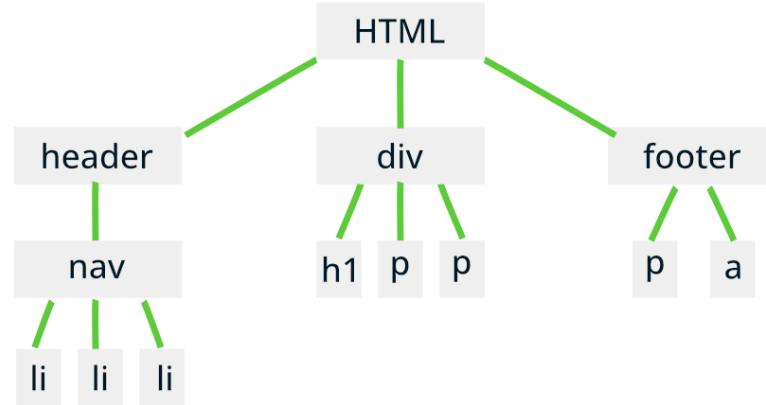
- **Traversing (Gezinme):** Bir öğeden başlayarak üst, alt veya kardeş öğelere ulaşmak
- **Cloning (Kopyalama):** Bir öğenin birebir kopyasını oluşturmak
- **Handling Dynamic Elements (Dinamik Öğeleri Yönetme):** Sayfaya sonradan eklenen öğelerle çalışmak

Traversing - DOM Ağacında Gezinme

Web sayfaları bir ağaç yapısı gibi organize edilir. Bazen belirli bir öğeyi seçmek için doğrudan id veya class kullanmak yeterli olmaz. İşte bu noktada **traversing** (gezinme) devreye girer.

Traversing ile bir elementin:

- Üst ögelerine (Ancestors)
- Alt Ögelerine (Descendants)
- Kardeş ögelerine (Siblings) erişim sağlayabiliriz.



jQuery Traversing - Ancestors (Üst Öğeler)

Ancestors (Üst öğeler), seçilen bir ögenin yukarı doğru hangi HTML elemanlarının içinde olduğunu belirlemek için kullanılır. jQuery'de bir ögenin üst öğelerini seçmek için aşağıdaki metodlar kullanılır:

Metod	Açıklama	Örnek Kullanım
.parent()	Seçilen ögenin doğrudan üst ögesini döndürür	<code>\$("p").parent();</code>
.parents()	Seçilen ögenin tüm üst öğelerini döndürür	<code>\$("p").parents();</code>
.parentsUntil(selector)	Belirtilen seçiciye kadar olan tüm üst öğeleri getirir	<code>\$("p").parentsUntil(".wrapper");</code>
.closest(selector)	En yakın belirtilen türdeki üst ögeyi getirir	<code>\$("p").closest("div");</code>

jQuery Traversing - Descendants (Alt Öğeler)

Descendants (Alt öğeler), seçili bir ögenin içinde bulunan diğer öğelere ulaşmak için kullanılır.

Metod	Açıklama	Örnek Kullanım
.children(selector)	Seçili ögenin doğrudan alt öğelerini getirir	<code>\$("div").children("p");</code>
.find(selector)	Seçili ögenin tüm alt öğelerini getirir	<code>\$("div").find("span");</code>

jQuery Traversing - Siblings (Kardeş Öğeler)

Siblings (Kardeş öğeler), aynı üst öğeye sahip olan öğelerdir. Bir ögenin kardeş öğelerini seçmek için kullanılan jQuery metodları şunlardır:

Metod	Açıklama	Örnek Kullanım
.siblings()	Seçili ögenin tüm kardeş öğelerini getirir	<code>\$("p").siblings();</code>
.next()	Seçili ögenin bir sonraki kardeş öğesini getirir	<code>\$("p").next();</code>
.nextAll()	Seçili ögenin kendinden sonraki tüm kardeş öğelerini getirir	<code>\$("p").nextAll();</code>
.prev()	Seçili ögenin bir önceki kardeş öğesini getirir	<code>\$("p").prev();</code>
.prevAll()	Seçili ögenin kendinden önceki tüm kardeş öğelerini getirir	<code>\$("p").prevAll();</code>
.nextUntil(selector)	Belirtilen öğeye kadar tüm sonraki kardeşleri getirir	<code>\$("p").nextUntil("h1");</code>
.prevUntil(selector)	Belirtilen öğeye kadar tüm önceki kardeşleri getirir	<code>\$("p").prevUntil(".header");</code>

jQuery Traversing - Filtering (Öğeleri Filtreleme)

Traversing işlemleri sırasında belirli öğeleri seçmek veya ayıklamak için filtreleme metodları kullanılır.

Metod	Açıklama	Örnek Kullanım
.first()	Seçili öğelerin ilkini getirir	<code>\$("li").first();</code>
.last()	Seçili öğelerin sonuncusunu getirir	<code>\$("li").last();</code>
.eq(index)	Seçili öğelerin belirtilen indeksindeki öğeyi getirir	<code>\$("li").eq(2);</code>
.filter(selector)	Belirtilen seçiciye uyan öğeleri getirir	<code>\$("li").filter(".selected");</code>
.not(selector)	Belirtilen seçiciye umayan öğeleri getirir	<code>\$("li").not(".disabled");</code>
.has(selector)	İçinde belirtilen öğeye sahip olanları seçer	<code>\$("div").has("p");</code>

jQuery Traversing - Tüm Metodlar

Metod	Örnek Kod	Açıklama
<code>add()</code>	<code>\$(“p”).add(“div”).css(“color”, “red”);</code>	<p> ve <div> öğelerinin rengini kırmızı yapar.
<code>addBack()</code>	<code>\$(“.container”).find(“.active”).addBack().css(“background”, “green”);</code>	.container ve içindeki .active öğelerinin arka planını yeşil yapar.
<code>andSelf()</code> (eskimiş)	<code>\$(“li.active”).next().andSelf().css(“background”, “yellow”);</code>	addBack() ile aynı işlevi sahiptir ancak eskimiştir.
<code>contents()</code>	<code>\$(“div”).contents().css(“border”, “1px solid blue”);</code>	<div> içindeki metin ve ögelere mavi çerçeve ekler.
<code>each()</code>	<code>\$(“li”).each(function(index, element) { console.log(index + “: ” + \$(element).text());});</code>	Tüm öğelerinin indeksini ve metnini konsola yazar.
<code>end()</code>	<code>\$(“div”).find(“p”).css(“color”, “red”).end().css(“border”, “1px solid black”);</code>	Once <p> öğelerini kırmızı yapar, sonra <div> öğelerine border ekler.

jQuery Traversing - Tüm Metodlar

Metod	Örnek Kod	Açıklama
has()	<code>\$(“div”).has(“p”).css(“border”, “2px solid green”);</code>	İçinde <code><p></code> bulunan <code><div></code> öğelerine yeşil çerçeve ekler.
is()	<code>if (\$("input").is(":checked")) { alert("Checkbox işaretelli!"); }</code>	Eğer <code><input></code> checkbox işaretliyse bir uyarı verir.
map()	<code>var texts = \$("li").map(function() { return \$(this).text(); }).get(); console.log(texts);</code>	Tüm <code></code> metinlerini bir diziye dönüştürüp konsola yazar.
not()	<code>\$("li").not(".active").css("opacity", "0.5");</code>	<code>.active</code> sınıfına sahip olmayan <code></code> öğelerinin saydamlığını azaltır.
offsetParent()	<code>console.log(\$(".span").offsetParent());</code>	<code></code> ögesinin en yakın konumlandırılmış üst ögesini konsola yazar.
slice()	<code>\$("li").slice(1, 3).css("background", "pink");</code>	İkinci ve üçüncü <code></code> öğelerinin arka planını pembe yapar.

Cloning - Elemanları Kopyalama

Bazen bir ögeyi tekrar tekrar kullanmamız gereklidir. Bunun için `clone()` metodunu kullanabiliriz. Kopyalama işlemi, özellikle formlar, dinamik listeler veya içerik çoğaltma gibi işlemler için idealdir.

Ancak, `.clone()` metodu sadece HTML içeriğini kopyalar, event'leri (olayları) **varsayılan olarak kopyalamaz**. Eğer event'lerin de kopyalanmasını istiyorsak `.clone(true)` kullanmalıyız.

```
$(“selector”).clone()
```

Cloning Kullanım Senaryoları

Cloning, özellikle dinamik form oluşturma, içerik çoğaltma ve tekrar eden işlemler için yaygın olarak kullanılır.

Kullanım Alanı	Açıklama
Dinamik Form Alanları	Kullanıcıların yeni giriş alanları oluşturmaya izin vermek
İçerik Çoğaltma	Tekrar eden içerik bloklarını oluşturmak
Şablon (Template) Kullanımı	Önceden tanımlanmış bir öğeyi tekrar kullanmak
Drag & Drop Uygulamaları	Öğeleri sürükleyerek kopyalama işlemlerinde

Ödev

Hedef: JSON veya XML formatında bir veri dosyası oluşturulmalı. Bu dosyada her ürünün adı, fiyatı, linki ve detay bilgisi yer almalı. Bu veriler kullanılarak sayfada ürünler dinamik bir şekilde jQuery ile listelenmeli. Ürünlere tıklandığında, o ürünün detay bilgilerini içeren bir pop-up açılmalı ve animasyonlar eklenerek etkileşim sağlanmalı.

Gereklilıklar:

- **JSON veya XML Dosyası Oluşturulmalı:** JSON veya XML formatında, ürün adı, fiyatı, linki ve detay bilgisini içeren bir dosya oluşturulmalı.
- **Dinamik Listeleme:** JSON/XML dosyasındaki veriler, jQuery ve AJAX kullanılarak alınmalı ve her ürün için dinamik olarak HTML öğeleri oluşturularak sayfaya eklenmelidir.
- **Animasyonlar:** Ürünlerin üzerine gelindiğinde bir animasyon eklenmeli (örneğin, renk değişimi veya boyut büyümesi gibi).
- **Pop-up Detayları:** Ürune tıklandığında, o ürünün JSON dosyasındaki detay bilgilerini içeren bir pop-up açılmalı.

Önemli Not:

HTML ve CSS kodları dışında tüm işlemler **sadece jQuery kullanılarak yapılmalı**. Ürünlerin dinamik olarak eklenmesi, animasyonların yapılması, pop-up açılması gibi işlemler **sadece jQuery ile gerçekleştirilmeli**. Ayrıca, **HTML ve CSS kodları da jQuery ile yazılmalı** ve tüm kodlar konsola yapıştırıldığında çalışması gerektiğini unutmayın. Hem HTML elementleri dinamik olarak oluşturulmalı hem de stil verilmelidir.