



# **CodeCraft**

## Bootcamp'25

# Manipulating HTML With jQuery & AJAX 2. Gün

# Konu Başlıklarısı

- jQuery ile HTML Nasıl Manipüle Edilir?
- **jQuery ile İçerik Değiştirme**
- **jQuery ile Öğeler Ekleme**
- **jQuery ile Öğeleri Silme**
- **jQuery css() - Stil Özelliklerini Değiştirme**
- **jQuery ile Boyutlar**
- **AJAX Nedir?**
- **AJAX & Sağladığı Avantajlar**
- **jQuery ile AJAX - Temel Yapı**
- **jQuery ile AJAX - Syntax ve Parametreler**
- **jQuery AJAX ve JSON**
- **jQuery AJAX Metodları ve Kullanımı**

# jQuery ile HTML Nasıl Manipüle Edilir?

Web sayfalarında içeriği değiştirmek, yeni elemanlar eklemek veya mevcut öğeleri kaldırmak için jQuery çeşitli yöntemler sunar. Bu sayede sayfayı yeniden yüklemeden dinamik içerikler oluşturabilirsiniz.

## Öne Çıkan Metotlar:

- `.html()`, `.text()`: HTML içeriğini değiştirme
- `.val()`: Form giriş değerlerini alma ve değiştirme
- `.append()`, `.prepend()`, `.after()`, `.before()`: Yeni içerik ekleme
- `.remove()`, `.empty()`: Elemanları kaldırma

# jQuery ile İçerik Değiştirme

- jQuery ile öğelerin içeriğini değiştirmek oldukça kolaydır. `html()`, `text()`, `val()` gibi metodlar ile içerik (`text`, `html` veya `form` verisi) üzerinde değişiklik yapılabilir. Bu metodlar, ögenin mevcut içeriğini almak ya da değiştirmek için kullanılır.
- **html()**: HTML içeriğini almak veya değiştirmek için kullanılır.
- **text()**: Sadece metin içeriğini almak veya değiştirmek için kullanılır.
- **val()**: Form elemanlarının (`input`, `select` vb.) değerini almak veya değiştirmek için kullanılır.

# jQuery ile Öğeler Ekleme

jQuery, dinamik olarak öğe eklemek için birçok metod sunar. `append()`, `prepend()`, `before()`, `after()` metodları, bir öğe eklerken hangi pozisyonda ekleneceğini belirlemenizi sağlar.

- **append()**: İçeriği ögenin sonuna ekler.
- **prepend()**: İçeriği ögenin başına ekler.
- **before()**: Ögenin öncesine içerik ekler.
- **after()**: Ögenin sonrasına içerik ekler.

# jQuery ile Öğeleri Silme

jQuery, bir öğeyi DOM'dan kaldırmak için `remove()` metodunu kullanır. Bu metod, öğeyi tamamen siler ve içerisindeki tüm alt elemanlar da silinir. `empty()` ise `remove()`'dan farklı olarak bir ögenin içeriğini temizler ama ögenin kendisini silmez.

Metod	İşlev	Kullanım Senaryosu
<code>remove()</code>	Seçilen öğeyi ve içindeki tüm öğeleri DOM'dan tamamen kaldırır.	Bir öğeyi tamamen kaldırmak istediğinizde kullanılır.
<code>empty()</code>	Ögenin içeriğini temizler ancak ögenin kendisini silmez.	Öge yerinde kalırken sadece içeriğinin silinmesi gereğinde kullanılır.

# jQuery css() - Stil Özelliklerini Değiştirme

jQuery ile öğelerin stilini değiştirmek için `css()` metodu kullanılır. Hem stil özelliklerini okumak hem de değiştirmek mümkündür.

```
css("propertynname","value");
```

```
css({"propertynname":"value","propertynname":"value",...});
```

# jQuery ile Boyutlar

jQuery, öğelerin boyutlarını ölçmek ve değiştirmek için çeşitli metodlar sunar. `width()`, `height()`, `innerWidth()`, `innerHeight()`, `outerWidth()`, ve `outerHeight()` metodları kullanılarak boyut bilgileri alınabilir veya değiştirebilir.

- **width() / height()**: Ögenin genişliğini ve yüksekliğini alır veya değiştirir.
- **innerWidth() / innerHeight()**: Ögenin iç genişliğini ve yüksekliğini (padding dahil) alır.
- **outerWidth() / outerHeight()**: Ögenin dış genişliğini ve yüksekliğini (padding ve border dahil) alır.

# AJAX Nedir?

- **AJAX**, web sayfalarının tamamen yeniden yüklenmesine gerek kalmadan, sunucudan veri almasını ve gönderilmesini sağlayan bir tekniktir. AJAX, aslında bir teknoloji değil, web sayfalarında dinamik içerik güncellemeleri yapabilen bir yöntemler setidir.
- Asenkron (asynchronous) yapısı sayesinde, web sayfasının arka planda veri alıp göndermesi sağlanır. Bu, kullanıcıların web sayfası ile etkileşimde bulunmaya devam ederken, sayfanın belirli bölümlerinin güncellenmesini mümkün kılar.
- AJAX, genellikle **JavaScript**, **XML**, **JSON** ve **HTML** kullanarak çalışır. Web sayfalarında içerik yenileme, form gönderme, liste güncelleme gibi işlemler AJAX ile daha hızlı ve kullanıcı dostu bir şekilde yapılır.

# AJAX & Sağladığı Avantajlar

- **Kesintisiz Kullanıcı Deneyimi:** Sayfa yenilemesi yapılmadığı için kullanıcılar işlem yapmaya devam edebilir.
- **Hızlı Tepkiler:** Sayfa tümüyle yüklenmeden, sadece gerekli veriler alınır, bu da işlem hızını artırır.
- **Verimli Veri Kullanımı:** Sadece gerekli veriler alınır ve gönderilir, böylece gereksiz veri transferi önlenir.

# jQuery ile AJAX - Temel Yapı

jQuery, AJAX işlemlerini daha basit ve daha az kodla yapmamıza olanak tanır. jQuery ile AJAX istekleri çok daha kısa ve anlaşılır hale gelir. jQuery'nin sunduğu `.ajax()`, `.get()`, `.post()` gibi fonksiyonlar, AJAX işlemlerini basitleştirir ve geliştiricilerin kodu daha verimli yazmalarını sağlar.

jQuery'nin AJAX fonksiyonları ile asenkron işlemler gerçekleştirebilir, veri gönderebilir ve alabiliriz. Bu fonksiyonlar, başarılı veya başarısız bir işlem durumunda çalışacak callback fonksiyonlarını da destekler.

# jQuery ile AJAX - Syntax ve Parametreler

```
$.ajax({
  url: "example.php",
  type: "POST",
  data: { username: "John", password: "12345" },
  dataType: "json",
  success: function(response) {
    console.log("Data received successfully:", response);
  },
  error: function(xhr, status, error) {
    console.log("An error occurred:", error);
  },
  complete: function(xhr, status) {
    console.log("Request completed, status:", status);
  },
  timeout: 5000,
  async: true,
  contentType: "application/json"
});
```

Parametre	Açıklama
url	AJAX isteğiinin yapılacak URL.
type	İstek türü (GET, POST, PUT, DELETE).
data	Sunucuya gönderilecek veri (nesne veya dize).
dataType	Sunucudan beklenen veri tipi (json, xml, html, text).
success	İstek başarılı olduğunda çalışacak fonksiyon.
error	İstek sırasında hata oluşursa çalışacak fonksiyon.
complete	İstek tamamlandığında çalışacak fonksiyon (başarı veya hata fark etmez).
timeout	İstek için belirlenen maksimum süre (ms cinsinden).
async	İsteğin asenkron olup olmayacağı (varsayılan: true).
contentType	Gönderilen verinin tipi (örneğin, application/json).
headers	Sunucuya gönderilecek özel başlıklar (örneğin, Authorization).

# jQuery AJAX ve JSON

**JSON (JavaScript Object Notation)**, verilerin yazılması ve okunması açısından oldukça hafif bir veri formatıdır ve AJAX ile iletişimde sıkça kullanılır. JSON, veri iletiminde XML'e göre daha hızlı ve daha verimli bir alternatifdir. JavaScript ile kolayca işlenebilir.

jQuery, JSON verisini almak, işlemek ve sunucuya JSON formatında veri göndermek konusunda oldukça güçlündür. JSON formatındaki veriler, JavaScript nesnelerine kolayca dönüştürülebilir.

# jQuery AJAX Metodları ve Kullanımı

jQuery, AJAX işlemlerini kolaylaştıran birçok metod sunar. Bu metodlar sayesinde sunucu ile veri alışverişi yapmak, form verilerini göndermek veya dinamik içerik yüklemek mümkün olur. Aşağıdaki tabloda jQuery'nin en yaygın kullanılan AJAX metodları ve işlevleri özetlenmiştir.

Metod	Açıklama	Kullanım Örneği
<code>\$.ajax()</code>	Genel AJAX isteği yapar, tüm HTTP metodlarıyla kullanılabilir.	<code>\$.ajax({ url: "data.json", method: "GET", success: function(response) { console.log(response); } });</code>
<code>\$.get()</code>	Sunucudan veri almak için kullanılan kısa yöntemdir.	<code>\$.get("data.json", function(response) { console.log(response); });</code>
<code>\$.post()</code>	Sunucuya veri göndermek için kullanılır.	<code>\$.post("server.php", { name: "Ahmet" }, function(response) { console.log(response); });</code>
<code>\$.getJSON()</code>	JSON formatında veri almayı kolaylaştırır.	<code>\$.getJSON("data.json", function(data) { console.log(data); });</code>
<code>\$.ajaxSetup()</code>	AJAX için varsayılan ayarları belirler.	<code>\$.ajaxSetup({ timeout: 5000 });</code>
<code>\$.ajaxPrefilter()</code>	AJAX isteği yapılmadan önce çalıştırılır.	<code>\$.ajaxPrefilter(function(options) { console.log("Ön işleme:", options.url); });</code>

# Ödev

**Hedef:** Bu ödevde, **JSON** veya **XML** dosyasından ürün verilerini **AJAX** kullanarak çekeceksiniz ve bu verileri dinamik olarak **ürün kartları** şeklinde ekranda listeleyeceksiniz. Her bir ürün kartı, ürün adı, fiyatı ve bir **dummy link** içerecek. "Ürünleri Listele" butonuna tıkladığınızda, ürünler ekranda listelenecek.

## Adımlar:

1. İlk olarak, bir **JSON** veya **XML** dosyası oluşturun. Bu dosyada her ürün için **name**, **price** ve **link** gibi bilgiler yer alacak.
2. Sayfanızda bir "Ürünleri Listele" butonu ekleyin.
3. Bu butona tıklandığında, oluşturduğunuz **JSON** veya **XML** dosyasından ürün verilerini **AJAX** ile çekin.
4. Veriyi aldıktan sonra, her bir ürün için dinamik olarak bir kart oluşturun. Kartlarda ürünün adı, fiyatı ve bir **link** olacak
5. Her ürün kartına, dosyanızdaki **link** bilgisine göre bir bağlantı ekleyin. Kullanıcılar bu kartlara tıklandığında, o ürünü ait **link** adresine yönlendirilecekler.