



CodeCraft

Bootcamp'25

jQuery Basics & XML / JSON

1. Gün

Konu Başlıkları

- jQuery Nedir? Kurulumu ve Kullanımı
- jQuery Syntax ve Selectors
- jQuery Event Handling
- XML Nedir?
- JSON Nedir?

jQuery Nedir?

- jQuery, JavaScript ile yazılmış, HTML belgelere kolayca erişim, manipülasyon, olay yönetimi ve animasyonlar eklemeyi sağlayan hafif, hızlı ve yaygın kullanılan bir kütüphanedir. Temelde HTML, CSS ve AJAX işlemlerini kolaylaştırır. Hızlı, hafif ve tarayıcı uyumludur.
- Google** ve **Microsoft** gibi büyük şirketler aktif olarak **jQuery**'yi kullanmaktadır.

jQuery Kurulumu Nasıl Yapılır?

- jQuery **3 farklı yöntem** ile kurulabilir:
 - CDN (En kolay yöntem)
 - Dosya İndirme (Offline kullanım için)
 - npm/yarn ile yükleme (Node.js projeleri için)



jQuery Syntax

jQuery'de temel yapı aşağıdaki gibidir. JavaScript'te olduğu gibi, ifadeler ";" (noktalı virgül) ile sonlandırılır

\$: jQuery'e erişmek için kullanılır.

Seçilen Nesne: Üzerinde işlem yapacağımız HTML etiketi

İşlem: Nesne üzerinde ne gibi bir eylem gerçekleştirebileceğini belirtir.

The diagram shows the jQuery syntax `$(Selector).action();`. Annotations explain each part: a red arrow points to the `$` sign with the text "Sign denotes jQuery function"; another red arrow points to the `Selector` with "Select the HTML element"; a blue arrow points to the `.action()` part with "Perform action on selected element"; and a red arrow points to the `.` character with ". separator".

`$(Selector).action();`

- \$. separator
- \$ Sign denotes jQuery function
- Select the HTML element
- Perform action on selected element

jQuery Temel Seçiciler

jQuery Selectors (Seçiciler), HTML sayfasındaki belirli elementleri seçmek için kullanılır. JavaScript'te `document.querySelector` veya `getElementById` gibi yöntemler yerine jQuery'nin güçlü ve kısa seçeneklerini kullanabiliriz.

jQuery'de Temel Seçici Kullanımı

Seçici	Açıklama	Örnek
*	Tüm HTML elemanlarını seçer	<code>\$(" * ")</code>
tagName	Belirtilen etiketi seçer	<code>\$("p")</code> (Tüm <code><p></code> etiketleri)
#id	ID'si belirtilen elemanı seçer	<code>\$("#myDiv")</code>
.class	Class'ı belirtilen elemanları seçer	<code>\$(".myClass")</code>

jQuery Filtreleme Seçicileri

Seçici	Açıklama	Örnek
<code>:first</code>	İlk öğeyi seçer	<code>\$("p:first")</code> (İlk <p>)
<code>:last</code>	Son öğeyi seçer	<code>\$("p:last")</code> (Son <p>)
<code>:even</code>	Çift index'li öğeleri seçer	<code>\$("tr:even")</code> (Çift satırlar)
<code>:odd</code>	Tek index'li öğeleri seçer	<code>\$("tr:odd")</code> (Tek satırlar)
<code>:nth-child(n)</code>	Belirtilen sıradaki öğeyi seçer	<code>\$("li:nth-child(3)")</code> (Üçüncü)
<code>:not(selector)</code>	Belirtilen seçici dışındaki öğeleri seçer	<code>\$("p:not(.exclude)")</code> (Class'ı exclude olmayan <p>'ler)

jQuery Selector Avantajları

Özellik	jQuery
Gizli elemanları seçme	<code>\$('div:hidden')</code>
:checked inputları bulma	<code>\$('input:checked')</code>
Son öğeyi (:last) seçme	<code>\$('li:last')</code>
Belirli bir metin içeren öğeleri (:contains()) seçme	<code>\$('p:contains("Merhaba")')</code>
Çift (:even) ve tek (:odd) satırları seçme	<code>\$('tr:odd')</code> veya <code>\$('tr:even')</code>

jQuery Event Handling

Event Handling (Olay Yönetimi), kullanıcıların web sayfasıyla **etkileşime girmesini** sağlar. jQuery, olayları (event) yönetmek için JavaScript'ten daha kolay ve okunabilir bir yöntem sunar.

Avantajları

Daha Kısa ve Okunaklı Kod → JavaScript'te uzun kod blokları gerektiren işlemler jQuery ile tek satırda yapılabilir.

Tarayıcı Uyumluluğu → jQuery, tüm tarayıcılarda tutarlı çalışır, JavaScript'te ise bazı tarayıcı farkları olabilir.

Daha Kolay Dinamik Element Yönetimi → `on()` metodu sayesinde sonradan eklenen elemanlara event bağlanabilir.

Event Delegation Desteği → Birden fazla öğeye event atamak yerine **ebeveyn element üzerinden yönetim sağlanabilir**.

Birden Fazla Event Tanımlayabilme → Tek bir öğe için birden fazla olay tek satırda yazılabilir.

jQuery Event Kullanımı

JavaScript'te olduğu gibi jQuery'de de olay yönetimi, HTML öğeleri üzerinde belirli etkileşimleri dinleyerek işlem yapmamızı sağlar. Kullanıcı bir butona tıkladığında, bir kutunun üzerine geldiğinde veya klavyeden bir tuşa bastığında bu olayları algılayıp tepki verebiliriz. JavaScript'te `addEventListener` ile yapılan işlemler, jQuery'de daha kısa ve okunaklı bir şekilde yazılabilir.

jQuery Event Tipleri

Her event'in bir türü vardır ve bunlar kullanıcının etkileşimine göre farklılık gösterir. Başlıca olay türleri şunlardır:

Olay Türü	Olaylar
Mouse Events	click, dblclick, mouseenter, mouseleave, mousemove, mouseover, mouseout
Keyboard Events	keydown, keypress, keyup
Form Events	submit, focus, blur, change, input
Window Events	load, resize, scroll, unload
Touch Events	touchstart, touchend, touchmove (mobil cihazlar için)

jQuery Event Objesi

Her olay tetiklendiğinde bir **event object** oluşur. Oluşturulan bu nesne, olay hakkında bilgiler içerir.

event.target: Olayın gerçekleştiği öğe.

event.type: Olayın türü (örneğin click, keydown).

event.pageX ve event.pageY: Olayın gerçekleştiği sayfanın koordinatları.

jQuery: on() Metodu ile Birden Fazla Olayı Yönetme

jQuery, bir öğeye birden fazla olayı tek satırda tanımlama avantajı sunar. JavaScript'te her olay için ayrı bir addEventListener kullanmamız gereklidir, jQuery'de on() metodu sayesinde birden fazla olay daha pratik bir şekilde atanabilir.

`$(selector).on(event,childSelector,data,function,map)`

Parametre	Açıklama
event (<i>Gerekli</i>)	Seçilen öğelere bağlanacak bir veya daha fazla olayı veya namespace'i belirtir. Birden fazla olay, boşluk ile ayrılmalıdır.
childSelector (<i>İsteğe Bağlı</i>)	Olay dinleyicisinin yalnızca belirtilen alt öğelere eklenmesini sağlar.
data (<i>İsteğe Bağlı</i>)	Olay tetiklendiğinde, fonksiyona iletilenek verileri belirtir.
function (<i>Gerekli</i>)	Olay gerçekleştiğinde çalışacak fonksiyonu belirtir.
map (<i>İsteğe Bağlı</i>)	{event: function, event: function, ...} şeklinde bir nesne kullanarak birden fazla olayı aynı anda atamayı sağlar.

jQuery: Dinamik Öğeler İçin on() Kullanımı

JavaScript'te olaylar genellikle **sayfa yüklenliğinde var olan öğeler** için atanır. Ancak, **sonradan eklenen öğelere olay eklemek için** farklı bir yöntem kullanmak gereklidir. JavaScript'te bu işlem için manuel olarak her yeni öğeye addEventListener atamamız gereklidir, jQuery'de on() metodu sayesinde bu işlem otomatik olarak yapılabilir.

```
$('.new-element').click(function () {
    alert("clicked");
});

$(document).on('click', '.new-element', function () {
    alert("clicked");
});
```

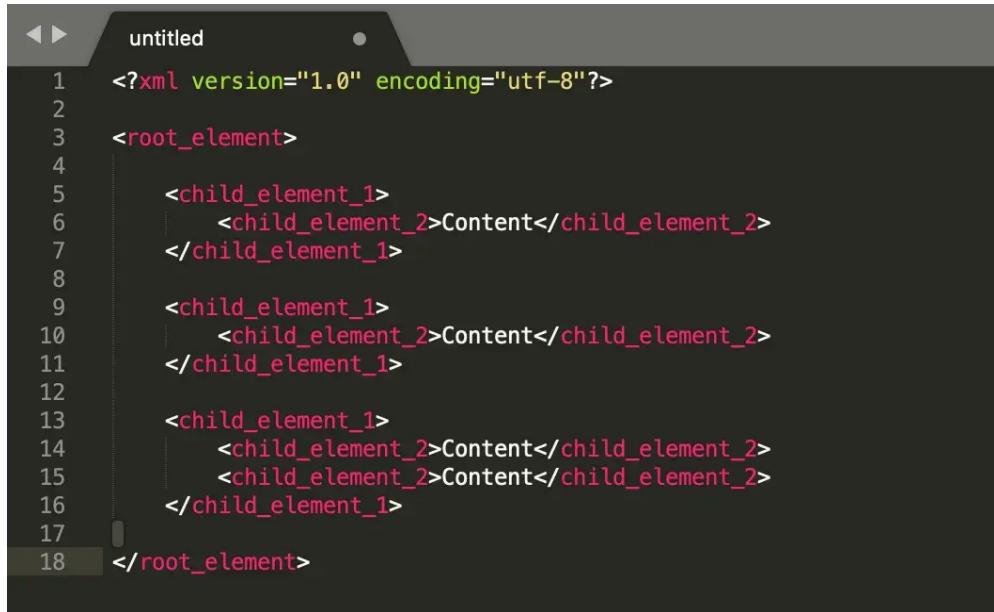
jQuery: one() Metodu

Bazı durumlarda bir olayın sadece **bir kez çalışması** istenebilir. Örneğin, bir butona basıldığında belirli bir işlemin sadece ilk tıklamada gerçekleşmesini ve sonrasında çalışmamasını isteyebiliriz. JavaScript'te bunun için ekstra kod yazmak gereklidirken, jQuery'de one() metodu ile bu işlem çok daha kolay hale gelir.

```
$("#btn").one("click", function() {  
    alert("This button can only be clicked once!");  
});
```

XML (eXtensible Markup Language) Nedir?

XML (eXtensible Markup Language), verileri saklamak ve taşımak için kullanılan bir işaretleme dilidir. HTML gibi bir yapıya sahiptir ancak **veri gösterimi** için kullanılır, yani tarayıcılarda sayfa oluşturmak yerine verileri düzenli bir şekilde saklar.

A screenshot of a code editor window titled "untitled". The code is written in XML and defines a hierarchical structure. It starts with a root element and contains three child elements. Each child element has two nested child elements, with the innermost being "Content".

```
<?xml version="1.0" encoding="utf-8"?>
<root_element>
  <child_element_1>
    <child_element_2>Content</child_element_2>
  </child_element_1>
  <child_element_1>
    <child_element_2>Content</child_element_2>
  </child_element_1>
  <child_element_1>
    <child_element_2>Content</child_element_2>
    <child_element_2>Content</child_element_2>
  </child_element_1>
</root_element>
```

JSON Nedir?

JSON (**JavaScript Object Notation**), veri saklamak ve taşımak için kullanılan hafif, metin tabanlı bir veri formatıdır.

JSON'un Öne Çıkan Özellikleri:

- **İnsan tarafından okunabilir** – Yapısı sade ve anlaşılırdır.
- **Makine tarafından işlenebilir** – Çoğu programlama dili JSON'u kolayca işleyebilir.
- **Anahtar-Değer Çiftleri Kullanır** – Veriler { "anahtar": "değer" } formatında saklanır.
- **Dil Bağımsızdır** – JSON, sadece JavaScript'e özgü değildir, birçok dil tarafından desteklenir.

JSON ve XML Karşılaştırması

Özellik	JSON	XML
Okunabilirlik	Daha sade ve kısa	Daha uzun ve karmaşık
Kullanım Kolaylığı	JavaScript ve diğer dillerde kolay işlenir	Daha fazla işleme gerektirir
Hız	Daha hızlıdır	Daha yavaştır
Veri Yapısı	Obje ve dizi temelli	Ağaç yapısı (etiket tabanlı)

Ödev

Hedef: Kullanıcı görev ekleyebilsin, silebilsin ve tamamlanan görevleri işaretleyebilsin.

Adımlar:

1. Kullanıcıdan görev metni almak için bir input alanı ve buton ekle
2. Butona tıklandığında yeni bir ekle
3. içindeki “Sil” butonuna tıklanınca öğeyi .remove() ile sil
4. Görevin tamamlandığını göstermek için .toggleClass("completed") kullan
5. Tamamlanan görevlerin tasarımları değişmeli.